

# The Project Management Mid-life Crisis

*A paper presented at the 2008 APM Conference*

# Contents

---

Contents.....	1
The Project Management Mid-life Crisis.....	2
Bibliography.....	8
Additional Resources.....	9
Interview: Pawel Brodzinski.....	11
Interview: Bruce P Henry.....	13
Interview: Alex S Brown.....	16
Interview: Johanna Rothman.....	19
Interview: Raven Young.....	21
Interview: Jim McCarthy.....	24
Jim McCarthy on The Core.....	26

# The Project Management Mid-life Crisis

by David Daly

## A Project Management Mid-life Crisis?

Over the last 10 years project management has come-of age. An increased awareness of best practice and the popularity of certification has seen a shift away from the adhoc approaches that used to be commonplace. Despite these improvements many of the same old problems are still being experienced:

- Spiralling project costs
- Schedule overruns
- Failure to deliver the required business benefits

During 2008 I conducted a series of interviews with six leading project management professionals and gurus. I discussed with them how project management has changed, what challenges we continue to face and how project management needs to progress into the future.

Each interview yielded a number of unique and fascinating insights into what is required to manage projects successfully. But across all of the interviews there was one consistent message: the way we manage projects has to improve. More startling however was the overwhelming sense that the way project management has evolved to date (i.e. an increased use of standardised processes) does not offer the solutions needed for the future. Perhaps project management has reached a mid-life crisis.

### What is a Mid-life Crisis?

My definition of a mid-life crisis is simple: "A point in time when you have come a long way and achieved a lot but you still don't feel you are where you need to be and you sense it is time to carve a new direction."

When I carefully reviewed all the interview material that I had gathered I was left with the feeling that this phrase was an apt way to describe the state of the project management profession. To understand why I felt this way it is necessary to consider in a little more detail who I interviewed and what they told me.

## Who Did I Interview?

This exercise was not a statistically sound survey of a typical cross-section of average project managers. In fact, as stated earlier, I interviewed only six people. They are all, however, recognised experts in a range of aspects of project management: from people and communication skills to advanced methods of risk management.

### Raven Young

Raven Young is a Seattle based project manager. She writes one of the most popular personal blogs on project management at [www.ravensbrain.com](http://www.ravensbrain.com). Raven began managing projects in the early 90s and now has over 14 years of experience in delivering hi-tech software and web projects for a range of companies.

### Pawel Brodzinski

Pawel Brodzinski has enjoyed a wide variety of roles and responsibilities in software development which has included extensive project management experience. He

“Across all of the interviews there was one consistent message: the way we manage projects has to improve.”

writes an interesting and informative blog about software project management at [blog.brodzinski.com](http://blog.brodzinski.com). Pawel currently works as the Chief Operating Officer for telecoms company Wind Mobile ([www.windmobile.pl](http://www.windmobile.pl)).

### Jim McCarthy

Jim McCarthy is somewhat of a legend in the world of software development. As one time director of the Microsoft Visual C++ Programme Management Team at Microsoft he has had firsthand experience of the challenges involved with managing large, bleeding edge software development projects. Since 1996 Jim has

devoted his professional career to researching groups and how they create products and organisations. Jim has written two books on team dynamics (McCarthy, 1995) (McCarthy & McCarthy, 2002) and publishes a wealth of information on his website [www.mccarthyshow.com](http://www.mccarthyshow.com).

#### Johanna Rothman

Johanna Rothman has written a number of popular books on project management and leadership skills including "Manage It" (Rothman, 2007) which won a 2008 Jolt Productivity Award. Johanna has been managing projects since the mid 80s and now runs her own project management consultancy: Rothman Consulting Group ([www.jrothman.com](http://www.jrothman.com)). She writes two fascinating blogs on managing product development (Rothman, Managing Product Development) and hiring technical people (Rothman, Hiring Technical People).

#### Alex S Brown

Alex S Brown is a Princeton graduate who has held a number of senior project management positions. These have included:

- Project manager at Chubb & Son
- Vice President, Senior Applications Manager at Merrill Lynch & Co
- Manager, Strategic Planning Office at Mitsui Sumitomo Insurance Group, USA

Alex now runs his own project management consultancy firm Real-Life Projects. Alex is also a popular speaker about project management in the USA as well as publishing numerous articles and other information on his website [www.alexsbrown.com](http://www.alexsbrown.com).

#### Bruce P Henry

Bruce P Henry has worked for a number of hi-tech companies including Microsoft, OpenDesign and Expedia (where he was their Senior Director of Quality Management and responsible for a team of about 100 software testers). In 2007 Bruce joined LiquidPlanner ([www.liquidplanner.com](http://www.liquidplanner.com)) which is a company that has built Web 2.0 based project management software. At LiquidPlanner Bruce is the brains behind the probabilistic scheduling engine that supports ranged estimation. His job title there is Director of Rocket Science (a tongue-in-cheek reference to his possession of a master's degree in physics). He blogs about project management and almost anything else at [www.brucehenry.com](http://www.brucehenry.com).

#### How has project management changed?

When I asked these people what has been the most significant change to the way we manage projects the responses were all different but shared a common

viewpoint: there has been an increase in awareness of the value and importance of project management as a discipline. Pawel Brodzinski summarised it best:

*"I think it's a change in awareness. We consider managing software projects as a more important part of software development than we used to. As an effect we have seen the development of things like the Agile Manifesto and other methodologies. We see an increasing role for project managers, analysts and architects which is a sign of moving several steps further down the road away from gung-ho software implementation."*

This was also backed up by Bruce P Henry when he said:

*"Thanks to folks like Fred Brooks, Steve McConnell and a host of others we have much more insight into what is driving the interactions inside of our projects. That insight has driven us to re-examine the way we run our software projects."*

**"I fear that the new standard teams are out of touch with the real practice of project management."**

#### The rise of process

Naturally this increased awareness of project management has led to the creation of formal project management methodologies as people have sought to understand and document best practice in this area. Alex S Brown is a member of several professional organisations and a keen contributor to the development of standards. However he had to confess that this is currently an area of frustration for him:

*"I think the Project Management Body of Knowledge (PMBOK) Guide is a great document but the recent versions suffer from a disconnect from reality. I fear that the new standard teams are out of touch with the real practice of project management."*

When I probed him about how this "disconnect from reality" has arisen he was keen to elucidate further:

*"Complex and long standards become so difficult to understand that you need an expert in the standard itself in order to use it and modify it. The more time the experts spend with*

*the standard itself, the less time they spend actually applying it and actually running projects. The standards then become more out-of-touch with real life. As they become more out-of-touch they also usually grow and become more complex. This cycle repeats thus making the standard become less and less useful over time."*

### The rise of certification

With processes and standards becoming harder to follow there has been a perceived need to be able to differentiate between those who understand them and those who don't. Certification programmes attempt to meet this need. Is this a good thing? Johanna Rothman and Raven Young weren't so sure. Both warned about the increasing popularity of certification amongst project managers and employers. Johanna Rothman described certification as a "red herring" and went on to say:

*"Gaining a Project Management Professional (PMP) qualification is a laudable goal. Anyone can learn something while studying for a PMP. But becoming a PMP is no guarantee of being able to manage a project. And the change I've seen is that companies are looking for PMPs but not helping their staff learn how to manage projects successfully."*

And Raven echoed this sentiment with the following comment:

*"I do fear that too much stress is placed on possessing a certificate rather than one's actual abilities to get the job done...I would encourage potential employers to look beyond certification to ensure you are getting a "whole PM", with all the necessary skills and experience to do the job."*

### What challenges are faced by project managers?

When I asked what was the greatest current challenge facing project management professionals I received a range of different responses.

#### Complexity

Bruce P Henry was clear about where he saw the greatest challenge: "In short, complexity" he replied. As someone working in software he was largely talking about *technological* complexity. But part of his answer, when he said "built by a geographically distributed team", showed that he was also referring to *organisational* complexity.

This is a current issue being faced by project managers in a wide variety of industries. Teams can be spread across the globe. Teams need to operate across multiple time zones.

Teams are expected to use matrix management, where a single individual is not necessarily assigned to a single project at a time. All of these organisational complexities, as Brice P Henry reminded me, "must be managed".

#### Communication

With project teams utilising increasingly complex organisational structures one thing that can become even more challenging than usual is effective communication across the whole of the project team. Alex S Brown was keen to underline to me that he sees effective communication as the most important challenge for project managers:

*"The greatest challenge is always communication. People blame project failures on missed dates, budget problems, technology or other issues. But ultimately a project failure is a failure to communicate. If we communicate well then schedules and budgets should be revised. If we communicate well then we should be able to overcome technology problems and disappointments."*

And Raven Young highlighted this as an important area too:

*"We send emails, conduct & facilitate meetings, distribute reports, metrics and schedules, send memos and notes, have chats in the hallway and ad-hoc meetings, but I believe PMs still aren't doing enough to communicate the right information at the right time to the proper audience and ensuring the message conveyed is both heard and understood. Sometimes we overload folks with information or bury key points or action items in long winded emails or speeches. I try to remember that communication is about more than speaking and writing. It also involves listening, understanding, confirming and sometimes restating, interpreting and adjusting your message to your audience and so much more!"*

#### Delivering late and over budget

Managing complexity and ensuring effective communication are, of course, important. But in many ways they are simply a means to an end. In this respect it was Pawel Brodzinski who really cut to the heart of the matter when he stated that "The challenge hasn't changed

**"The greatest challenge is always communication."**

over the years: projects are still being delivered late and over budget.”

Unfortunately you do not have to look very far to find evidence that supports Pawel Brodzinski’s observation. In 2007 Joe Harley, Chief Information Officer at the Department for Work and Pensions, estimated that “only 30 percent” of government IT projects and programmes are successful (Branigan, 2007).

At the end of 2007 Tata Consultancy Services released research based on 800 interviews with middle and senior IT managers. Amongst the conclusions reached were that 62% of organisations experience projects that are delivered late and 49% experience budget overruns (Dynamic Markets Ltd for Tata Consultancy Services, 2007).

In April 2008 a report prepared by PricewaterhouseCoopers into the 30 most important European transport projects predicted that they will overspend by €40billion (PropertyWeek.com, 2008). What’s more, the same report blamed the overspend, at least in part, on “poor project management”.

#### Is it time to carve a new direction?

It is clear that, despite the rise of process and certification, the project management profession is still failing to reliably deliver success. So what do we need to change? Is improving and refining our existing processes the answer? Will it help if even more project managers gain certification?

There can be no doubt that project management processes, when applied wisely, can deliver benefits. Similarly it is beyond question that certification programmes for project managers have done much to improve many people’s knowledge of project management techniques.

“Team = product. All the virtues and vices of the team express in the product, and vice-versa.”

However, when asked how project management needed to evolve into the future, not one person said that processes or certification would provide the answer. The changes we have seen over the last 10 years, although worthwhile, will not provide the solutions of the future. It is time to carve a new direction for the project management profession.

#### The direction we need to carve

Throughout the interviews I collected numerous practical tips for how projects could be managed more effectively as well as predictions and recommendations for how project management should move forwards as a profession. I have summarised this advice by grouping it together into three key areas:

- Focussing more on people
- Concentrating on delivering value
- Moving away from following a single project management process

#### Focussing more on people

Back in 1995 Jim McCarthy wrote a book entitled *Dynamics of Software Development* (McCarthy, 1995) which contained 54 “rules” for software development. When I asked him which rule he felt was still most relevant today he was quick to reply:

*“Team = product. All the virtues and vices of the team express in the product, and vice-versa.”*

“Team = product” is an eloquent yet simple way of describing the fact that the quality of a product is entirely dependent on the quality of the team producing that product. Not only that, but if you find aspects of the product that are defective, then you must first look to fix those defects in the team after which an improved product will then naturally follow.

Whilst Jim McCarthy was perhaps the most evangelical in this area others offered similar views on the importance of teamwork. Asked about what to look for when bringing someone on to a project team Raven Young replied:

*“If you already have a team in place that you are adding to then the most important quality to look for is someone who will be a good fit with respect to team dynamics. You want an individual who will enhance the current team’s skills and make everyone better, not a disruptor. Qualities like drive, initiative, good communication and collaboration, flexibility and adaptability are all important but inconsequential in the end if the team doesn’t gel.”*

And Bruce P Henry commented that:

*“Projects, not just software projects, are fundamentally social in nature. The old way of viewing projects (particularly in software) as task lists executed by “resources” against a fixed and certain timeline is dead and should be buried.”*

It is interesting to note that the theory of designing high performing teams and of the necessity for ensuring that they have a complimentary blend of skills and experience is nothing new. Back in 1981 Raymond Meredith Belbin

published his seminal findings on what roles people have to fill in order to form an effective team in his book *Management Teams* (Belbin, 2003). But from the answers I received it is clear that we have still not reached a stage where these ideas are being widely applied in practice. It is equally clear that this must change.

### Concentrating on delivering value

What is the purpose of any project? Is it to deliver a product that meets defined requirements? Is it to ensure that defined outcomes are delivered on time and within budget. What about controlling risk? What about ensuring appropriate levels of quality?

All of these descriptions of purpose are plausible. However they all fail to address the fundamental reason why any project is undertaken: to deliver value to an organisation. Raven Young cited what Eric D Brown said in his article about Information Technology Leadership and Alignment (Brown, 2008):

*“Projects must fit into your strategic goals and must deliver value to your organisation.”*

Most organisations have more projects that they would like to deliver than they are able to deliver (due to resource and other constraints). In this context, how they choose which project to deliver and when, becomes crucially important. Project Portfolio Management (PPM) provides an answer in this area as described by Raven Young:

*“That’s exactly what PPM is all about: working on high value projects, as defined by the organization’s goals, to get the best return and ensure the PM and team are working on only approved, value-aligned projects. Besides, if you don’t prioritise the projects in your portfolio, how can you know what to work on next? Which brings the most revenue or provides the bigger impact to the customer?”*

The importance of PPM and tying project delivery to organizational goals was also recognised by Alex S Brown:

*“I expect that there will be tremendous discussion around how to integrate project management into the whole enterprise. Programme management and project portfolio management will become increasingly important topics in the next ten years.”*

It is evident that PPM offers solutions in this area but I would contend that it is not yet being applied either widely enough or at a fine enough level of granularity. Many of today’s projects have the potential to be sub-divided into smaller, independent projects which could then be managed as a portfolio. This has the benefit of enabling an assessment of which of these smaller projects deliver the

most value so that they can be implemented first. What’s more, if priorities change or new requirements emerge, then this approach provides the built-in agility and flexibility to accommodate them.

In fact, PPM is a great way of providing agility and flexibility and it is clear that many modern approaches to managing projects (like, for example, Extreme Programming) are actually implementations of the PPM concept: they simply provide ways to prioritise what to do next based on what is most valuable to the organisation.

### Moving away from a single process

Many companies and organisations have concluded that to manage projects successfully it is necessary to have a defined, company-wide project management process in place. Furthermore they invest time and money ensuring that all their project managers follow that process. Why is this? Johanna Rothman offered one explanation:

*“Senior managers want to see results out of IT. But most senior IT managers (CIOs and their counterparts) don’t know enough about how to measure projects or about how to think about project success. So they ask their Project Management Offices (PMOs) to mandate a single process for all projects.”*

But this one-size-fits-all approach is not a panacea. Whilst it may seem tempting because of its simplicity it actually does more harm than good. Johanna Rothman stated that those who choose this option “hamstringing their project managers”. She continued by explaining that to deliver success project managers must be able to choose

“To deliver success project managers must be able to choose the right processes, tools and techniques.”

the right processes, tools and techniques based on a given project’s context.

Project management processes can provide a useful tool-set for project managers to tap into. They also provide a means for people to think about, analyse and discuss the different ways that projects can be managed. But the blind application of any single process, no matter how well designed, does not constitute good project management.

### The Project Management Mid-life Crisis

Having looked at who I interviewed and what they had to say; having considered their thoughts on how project

management has changed, the challenges we still face and the areas that we need to improve; having covered all of this I would now like to return back to my simple definition of a mid-life crisis: “A point in time when you have come a long way and achieved a lot but you still don’t feel you are where you need to be and you sense it is time to carve a new direction.”

Has the project management profession come a long way and achieved a lot? Without question this is true. We have seen an increased awareness of project management and an accompanying rise in process and certification.

Is the project management profession where it needs to be? Clearly not. Too many challenges are still being experienced including complexity, effective communication and, most crucially of all, too many projects being delivered late and over budget.

The changes of the last 10 years will not provide the answers we need for the next decade: it is time to carve a

new direction for project management. The three key areas that require attention are:

- Focussing more on people
- Concentrating on delivering value
- Moving away from following a single project management process

Given these facts it is true to say that the project management profession as a whole has reached a mid-life crisis. In some ways this is not a bad thing. For many people the phrase “mid-life crisis” is an intrinsically negative term. It conjures up images of middle-aged men wearing ill-fitting leather jackets. But, in fact, a mid-life crisis can be a moment of rebirth: a catalyst for positive change. Going through a mid-life crisis may not be a pleasant experience in itself but it has the potential to usher in a new era that is not only productive and rewarding but also truly meaningful.

*Do you agree that project management has reached a mid-life crisis and that it is time to focus more on people, concentrate on delivering value and move away from following a single project management process? Or do you think that other solutions will provide the best way forwards? Either way I'd love to hear your views. You can join the debate either by emailing me at [david@daly.net](mailto:david@daly.net) or logging on to <http://outofthetriangle.wordpress.com/>*

“The changes of the last 10 years will not provide the answers we need for the next decade: it is time to carve a new direction for project management.”

# Bibliography

---

Belbin, R. M. (2003). *Management Teams: Why They Succeed or Fail* (2nd ed.). Butterworth-Heinemann.

Branigan, T. (2007, May 18). *Most government IT projects 'not successful'*. Retrieved 2008, from The Guardian: <http://www.guardian.co.uk/business/2007/may/18/politics.society>

Brown, E. D. (2008, June 2). *Information Technology Leadership & Alignment*. Retrieved 2008, from Aligning Technology, Strategy, People & Projects: <http://ericbrown.com/information-technology-leadership-alignment.htm>

Dynamic Markets Ltd for Tata Consultancy Services. (2007, August). *IT Projects: Experience Certainty*. Retrieved 2008, from [http://www.tcs.com/thought\\_leadership/Documents/independant\\_markets\\_research\\_report.pdf](http://www.tcs.com/thought_leadership/Documents/independant_markets_research_report.pdf)

McCarthy, J. (1995). *Dynamics of Software Development*. Microsoft Press.

McCarthy, J., & McCarthy, M. (2002). *Software for Your Head*. Addison Wesley.

PropertyWeek.com. (2008, April 8). *EU transport schemes set for €40bn overrun*. Retrieved 2008, from PropertyWeek.com: <http://www.propertyweek.com/story.asp?sectioncode=36&storycode=3110618>

Rothman, J. (n.d.). *Hiring Technical People*. Retrieved from <http://jrothman.com/blog/http/>

Rothman, J. (2007). *Manage It!: Your Guide to Modern, Pragmatic Project Management*. Pragmatic Bookshelf.

Rothman, J. (n.d.). *Managing Product Development*. Retrieved from <http://jrothman.com/blog/mpd/>

# Additional Resources

---

## *The Interviews*

All the interviews and an electronic version of this paper are available for download from:

<http://outofthetriangle.wordpress.com/>

Further information about each interviewee and their thoughts on project management is available from the following web sites:

Pawel Brodzinski:	<a href="http://blog.brodzinski.com">blog.brodzinski.com</a>
Bruce P Henry:	<a href="http://www.brucehenry.com">www.brucehenry.com</a>
Alex S Brown:	<a href="http://www.alexsbrown.com">www.alexsbrown.com</a>
Johanna Rothman:	<a href="http://www.jrothman.com">www.jrothman.com</a>
Raven Young:	<a href="http://www.ravensbrain.com">www.ravensbrain.com</a>
Jim McCarthy:	<a href="http://www.mccarthyshow.com">www.mccarthyshow.com</a>

## *Other websites about project management, leadership and people skills*

### **10x Software Development**

<http://forums.construx.com/blogs/stevemcc/default.aspx>

Advice and opinions from legendary expert on managing software development Steve McConnell.

### **A Girl's Guide to Project Management**

<http://www.pm4girls.elizabeth-harrin.com/>

Informative thoughts on project management from Elizabeth Harrin.

### **Agile Project Planning**

<http://www.extremeplanner.com/blog/>

Information and insights about agile project planning and software development by Dave Churchville.

### **Aligning Technology, Strategy, People & Projects**

<http://ericbrown.com/>

Thoughts from Eric D Brown on aligning technology, strategy, people and projects.

### **Construx**

<http://www.construx.com/>

The homepage for Construx Software, Steve McConnell's software consultancy, this site has many useful resources including white papers, estimation tools and recommended reading lists.

### **IT Leadership**

<http://blogs.techrepublic.com.com/tech-manager/>

Information about IT leadership from TechRepublic (which also has many other useful IT related resources).

### **PickTheBrain**

<http://www.pickthebrain.com/>

A website dedicated to self improvement with a focus on personal productivity, motivation, positive psychology and self education.

### **PM Hut**

<http://www.pmhut.com/>

A large collection of categorized articles on project management.

### **Practicing Earl**

<http://forums.construx.com/blogs/earl/default.aspx>

Thoughts from Earl Beede on product development (especially software).

**Project Management 2.0**

<http://www.wrike.com/projectmanagement>

Views and analysis of how Enterprise 2.0 technologies affect contemporary project management and what impact they have on an organisation as a whole.

**Project Shrink**

<http://blog.softwareprojects.org/>

Blog by Bas de Baar that focuses on the importance of people in software projects.

**Undocumented Features**

<http://www.undocumentedfeatures.com/>

Stacey Douglas presents her ideas about software project management, business analysis and improving the software development lifecycle.

**The Practice of Leadership**

<http://www.thepracticeofleadership.net/>

Articles by George Ambler on leadership skills.

**Tyner Blain**

<http://tynerblain.com/>

Scott Sehlhorst's thoughts on the most effective ways to develop software.

## *Recommended Reading*

**Coaching For Performance: Growing People, Performance and Purpose**

*by Sir John Whitmore (Nicholas Brealey Publishing Ltd)*

This book looks at coaching methods and, by comparing them to more traditional management approaches, shows how and why they can be used to get the best out of people.

**Dealing with People You Can't Stand: How to Bring Out the Best in People at Their Worst**

*by Dr. Rick Brinkman and Dr. Rick Kirschner (McGraw-Hill Professional)*

Despite the somewhat negative title this is a useful book that covers the psychology of what makes people behave the way they do and provides practical tips for how to get the best out of them.

**Dynamics of Software Development**

*by Jim McCarthy (Microsoft Press)*

One of the first works to really focus on the people-centric nature of software development this book still contains many insights that are equally relevant today.

**First, Break All The Rules**

*by Marcus Buckingham and Curt Coffman (Pocket Books)*

An excellent book which describes (in a readable fashion) the results and conclusions of research into how the world's greatest managers get the best out of their people.

**Rapid Development: Taming Wild Software Schedules**

*by Steve McConnell (Microsoft Press)*

Seminal work by world renowned author on software development practices. This book is now a little dated but the fundamental principles that it covers remain as salient as ever.

# Interview: Pawel Brodzinski

13<sup>th</sup> February 2008

Over the coming weeks I will be publishing a series of interviews with successful project management professionals, experts and gurus. My main aim for this series is to discover and publish a range of views and opinions on what challenges project managers face, what approaches they find useful and how they perceive the future of project management.

This week I spoke to Pawel Brodzinski who has had a wide range of experiences within software project management and often writes entertainingly about them on his blog at [blog.brodzinski.com](http://blog.brodzinski.com).

**Q: Many people may already know you from your blog on software project management. Can you give us a brief summary of your career to date?**

My career so far has been quite a mixture of different positions related to software development including both “rank and file” and management roles. I worked at every stage of the software development lifecycle before moving to management roles which I think gives me quite a unique background in the software industry. For a time I worked for a large company building from scratch a development and implementation team for a new ERP software product.

“What I find is that everything starts with getting people involved.”

Unfortunately I couldn’t stand corporate politics and decided to move to a small company where I took over the development and project management division. That company split up and I ended up as COO of a spin-off from it: Wind Mobile.

**Q: What are you currently working on?**

Apart from fire fighting on projects whenever needed I look after an ongoing process of rebuilding the technical team within the company. The original goal was to move the company from a “guerrilla” operating model on to something much more organized. After relocating the company, a significant “refresh” of the team and re-implementing a project management methodology along with serious improvements in product quality it is now

time to improve the software development part of the business.

**Q: Wow! Sounds hectic. Do you find it difficult to manage your multiple commitments?**

Maybe it sounds hectic but it isn’t so hard. Bear in mind that what I am describing is more than just a couple of months work! I just try to find the biggest issue at any one time and fix it. This cycle is executed again and again because there are always things which can be improved.

**Q: Do you use any specific time management, organisation or personal productivity techniques to make sure you get everything done that you need to?**

If you try to do too many things at the same time then you will fail. What I find is that everything starts with getting people involved – I don’t even try do everything personally. Having a great team (and I do have one!) makes things much easier than they at first look. Of course having a specific type of character helps, as you have to deal with quite a lot of stress and be highly committed, but that’s pretty obvious. I don’t think a specific technique would help though if your character doesn’t suit the job.

**Q: What do you see as the greatest current challenge facing software project managers and what do you personally do to overcome it?**

I think the challenge hasn’t changed over the years: projects are still being delivered late and over budget. Changing this is the key challenge. Personally I don’t believe that simply using any known methodology is an answer here. Software projects differ so much that I don’t think there will be any universal solution anytime soon. The only thing you can do is to try to find your own solutions for the teams and clients you have. I’m far from being an evangelist for any single methodology. Instead I constantly try to find and apply useful techniques from any of them. That ongoing improvement process is my solution to the challenge.

**Q: What do you think has been the most important change in software project management over the last 10 years?**

I think it’s a change in awareness. We consider managing software projects as a more important part of software development than we used to. As an effect we have seen

the development of things like the Agile Manifesto and other methodologies. We see an increasing role for project managers, analysts and architects which is a sign of moving several steps further down the road away from gung-ho software implementation. I don't consider Agile itself as the most important change as this is an answer to a specific group of problems only. I consider Agile to be a side-effect of the real change in awareness that has happened in executives' minds.

**Q: What is your number one PM tip, trick or technique?**

Communicating. Every time there is an issue to resolve I will talk with all the interested parties, understand the root of the problem and find a way forwards. Very often you don't need to dig very deeply to find a solution. You will find that people just aren't talking to each other or trying to understand different points of view. They need a person, a proxy if you like, who will forward the right pieces of information to the right endpoints. When people see me running from one room to another, asking hundreds of questions and making hundreds of phone calls they know I'm probably trying to help people to communicate!

**Q: What characteristics do you look for when bringing people on to your team?**

Commitment, a willingness to learn and an ability to be a good team player. If you've got those three, the rest (including knowledge) can easily be built over time. Every issue I've ever had with a person on my team was because they sucked in one of those areas.

**Q: That's interesting. I have found that a certain technical aptitude/ability to think logically is important for technical roles. Does that mean you don't include any technical tests when hiring new recruits for development positions?**

Over time I ask more and more technical questions. I need to evaluate the candidate's knowledge to decide if they're worth the money they expect. However I usually can accept either inexperienced or experienced people as

long as they have the right attitude. The characteristics I have mentioned tell me whether I want to hire a person at

**“I consider Agile to be a side-effect of the real change in awareness that has happened in executives' minds.”**

all whereas technical questions point me to where a candidate could fit into my team.

**Q: Whilst we're on the subject of technical skills, how much programming knowledge do you think someone managing a software project has to have?**

You can have no programming knowledge and still be a great software project manager. On the other hand some programming knowledge definitely helps. I like to see at least a basic experience of development but it is not a must-have for me.

**Q: How do you see software project management evolving into the future?**

I don't think there's a big break-through ahead. Software projects have a long history of being over time and over budget despite numerous new methodologies and techniques being developed. That's why a revolution shouldn't be expected. Instead I think we will all rather slowly move forwards by improving our performance and adjusting the techniques we use. Gradually we will see a few more projects completed within their planned constraints.

**Q: Thanks for taking the time to share your thoughts. Is there anything else you would like to add?**

Just to say thanks for the interview and best wishes to all readers of Outside of the Triangle!

# Interview: Bruce P Henry

20<sup>th</sup> February 2008

This week I caught up with Bruce P. Henry, Director of Rocket Science at LiquidPlanner, to find out what he's been up to, glean a little more information about LiquidPlanner and explore his views on the future of project management.

**Q: Many people will already know you from your blog *Bruce's Brain* and your involvement with LiquidPlanner. Can you give us a brief summary of your career to date?**

My involvement with technology started in Hollywood. I ran a computerised teleprompting business out of my

“I think working at a good start-up is kinda like riding a hyperactive roller coaster.”

garage for a couple of years before switching to Macintosh consulting. After a year or so of 18-hour days, I left Los Angeles and moved to Laramie, Wyoming to pursue a degree in Physics. I took my BS in Physics and BS in Mathematics from the University of Wyoming and then headed to graduate school at the University of Washington. With my MS in Physics firmly in hand, I began testing software for Microsoft as a contractor. I went full-time at Microsoft but was lured away by Ed Jung and Nathan Myhrvold to join OpenDesign. Two years later I joined Expedia as a test lead and, over five years, worked my way up to becoming their Senior Director of Quality Management. I left Expedia and joined Charles and Jason at LiquidPlanner in 2007.

**Q: I imagine LiquidPlanner has been the focus of your attention for the past year or so. How has that time been?**

I think working at a good start-up is kinda like riding a hyperactive roller coaster. You have these huge ups and downs and sometimes they come just minutes apart. In general working at LiquidPlanner has been fantastic. I've gotten to do everything from writing some code (not much, but some) to meeting with investors and the press.

**Q: Can you tell us about some of those ups and downs?**

I'd say the real high points were figuring out uncertainty and launching at the DEMO conference. The “Aha!” moment when we figured out how to capture and treat uncertainty and first saw our models behaving like real projects was something else. Literally the hair stood up on the back of my neck when I saw the sheer magnitude of what we were uncovering. There is so much opportunity for improvement in project management. Similarly launching at DEMO: standing up on that stage and just laying everything on the line and then hearing that people “got it”. That was amazing and really gratifying.

The low points for me have been realising the sheer scale of the work we have in front of us and cutting beloved features in order to make our launch schedule. There is such a huge amount of stuff we could be working on that choosing just one or two is really hard. I always find myself second guessing our choices and asking “Is this really the most

important thing we could be adding to the product?”. Cutting features to make our launch date was pretty painful, even knowing that in the end it was the right thing to do.

**Q: What specifically are you currently working on?**

I am:

- Building demonstration videos for the product
- Writing help content
- Writing specifications for upcoming enhancements
- Travelling to faraway lands (Austin, Mountain View, and Penn State) to spread the word
- Building and integrating the analysis tools for our website traffic and product usage metrics
- Consulting with our CEO on VC and pricing strategy
- Keeping the fridge filled with soda
- Blogging

**Q: What techniques have you used for the development of LiquidPlanner?**

We use a “kinda scrum” methodology. I say “kinda” because we don't do things like daily meetings formally.

But we're all pretty much in one room so daily meetings happen whether they're formal or not. We built all of our UI prototypes in PowerPoint believe it or not. PowerPoint turned out to be really great because we could iterate on a design in a matter of hours and really look at the result. There were well over 200 versions of the PowerPoint prototype and they were worth every minute we spent on them.

**Q: Are you using the LiquidPlanner yet to manage further development?**

We have been using LiquidPlanner internally since early in 2007. We cut our first lines of code back in February of 2007 and began using the tool to plan our work in early April. It was pretty primitive back then but we got enormous value out of that experience. We use it religiously today for everything from tracking our bugs to planning our marketing. We are always using at least one version ahead of what we've released.

**Q: Can you tell us a little more about probabilistic scheduling and the LiquidPlanner scheduling engine?**

The term "Probabilistic Scheduling" was actually born fairly late in our process. When we first started out we knew that we wanted to build estimates in ranges. We had been to a Construx estimating training course and were completely convinced that ranged estimates were the way to go. But there were no tools that we could find that supported them. Sure, there were some that did the PERT thing with best/expected/worst case. But they just boiled those down into one number and that lost all the richness that we felt was there.

We looked at ranges and said "Well this is really an expression of probability". So we treated it like a probability. And then something magic happened. I had this model in Excel and when we got the thing working we got all these crazy answers out...

The amazing thing was that the answers seemed to be telling us what we'd always known as project managers. Like a project with 10 people working on 10 tasks of about one month duration will take more than a month to finish.

Yeah, like duh. Every project manager knows that.

But the tools we're used to using don't seem to get it. Try that schedule in MS Project and it'll tell you that you'll be done in one month. Then you'll have to add a buffer and argue with the project sponsor about why the buffer is so big (or there at all).

But sitting there in front of us was an incontrovertible mathematical reason why this happens.

I was all like, "Oh yeah... that makes sense." and then I was bouncing off the walls excited for a week. Realising

how much work it was going to take to do the calculations in a rigorous way, in real time, for projects with hundreds or thousands of tasks and a buncha people sobered me up pretty quick though.

Anyway, we basically take the range and infer a probability distribution. Then we use those distributions and the relationships between the people and the tasks to calculate a distribution for the project as a whole. Simple. The scheduling engine itself runs in the background on our servers while the user continues to work on other things in LiquidPlanner.

**Q: What do you see as the greatest current challenge facing software project managers and what approaches do you think are required to address it?**

We not only expect the code to work but we want it pretty, with shiny buttons and drop shadows, and running asynchronously on a distributed network with storage "in the cloud" and servers that repurpose themselves on the fly as load patterns shift and built by a geographically distributed team so it is cheap. And we want it yesterday.

In short, complexity. That's what I think is our biggest challenge. We're still developing on the bleeding edge for the most part and that's a good thing but the complexity must be managed.

To manage complexity software project managers must walk a narrow line between being so risk averse that they deliver boring obsolete software and burying the project under a mountain of failures. Complexity and newness introduce risk. And with risk comes failure, always. As project managers we need to be able to be open and honest about the risks and take them in intelligent ways. That also means that we need to have a pretty good understanding of the underlying technologies so we can explain them to our non-technical stakeholders.

"The amazing thing was that the answers seemed to be telling us what we'd always known as project managers."

**Q: What do you think has been the most important change in software project management over the last 10 years?**

Wow. This is a hard question. It is really hard to tell what is most important until it is a few years behind you.

I think we've come to understand a lot more about the dynamics of software development projects. Thanks to

folks like Fred Brooks, Steve McConnell and a host of others we have much more insight into what is driving the interactions inside of our projects. That insight has driven us to re-examine the way we run our software projects.

So, I'd say that agile methodologies and the new culture of openness in software development will, with hindsight, be the most important. I think that's because we're starting to treat the people involved more like people and beginning to acknowledge their human motivations in our software development processes.

**Q: What is your number one PM tip, trick or technique?**

Get your estimates in ranges.

I'm not kidding. Even if you do nothing "smart" with the range it gets you in the mindset of thinking of your project plan as a forecast rather than a road map. Once that

“Embracing uncertainty will revolutionise project management.”

happens you can start discussing what ifs, best cases and worst cases in a constructive atmosphere. It completely changes the nature of the interactions between the people, be they developers, testers, stakeholders or whoever else.

**Q: What characteristics do you look for when bringing people on to your team?**

People who are smart, passionate, and get things done.

**Q: How much programming knowledge do you think someone managing a software project has to have?**

The smaller the team the more knowledge they need. But I think that to be a truly great software project manager you need to have a passion for software. You have to care about the thing that you're building and

relate to the people who are building it. People with a passion for software will naturally have some knowledge of programming.

I think of asking what programming languages a person knows as a quick test for actual software passion.

**Q: What is your biggest bug bear/gripe/pet peeve?**

It drives me crazy when people say they want a different outcome in their projects but they don't want to change their behaviour.

**Q: Indeed. I seem to remember a quote to the effect of "only a fool does the same thing and expects different results". How do you see software project management evolving into the future?**

Software development will become even more distributed than it is today. More of the social aspect of software development projects will begin to surface in our tool set. These social aspects will begin to let us see how people fit into our projects and how the people affect outcomes. Software, in the end, is a human-centric endeavour. Software is built by people, for people. It just happens to run on a machine and that often clouds our perception of what is important in software.

Software project management will have to evolve with that in mind. We, as project managers, will have to embrace the variability and uncertainty that comes from having humans "in the loop". To help us do that, our tools will improve to enable more natural, healthy communication between all the members of a far-flung team. Projects, not just software projects, are fundamentally social in nature. The old way of viewing projects (particularly in software) as task lists executed by "resources" against a fixed and certain timeline is dead and should be buried.

Embracing uncertainty will revolutionise project management just as quantum mechanics and the embracing of uncertainty revolutionised physics.

# Interview: Alex S Brown

27<sup>th</sup> February 2008

This week I had the pleasure of asking Alex S. Brown to share some of his extensive project management knowledge and experience with us. Alex has worked for a whole host of large companies in a variety of environments. He now runs his own business delivering training and coaching to other project management professionals as well as being a popular speaker at numerous events.

**Q: Many people may already know you from the articles you publish on [www.alexbrown.com](http://www.alexbrown.com) or through your training and speaking work. Can you give us a brief summary of your career to date?**

I graduated from Princeton University and my first experience with project management came at Automatic Data Processing. I did not even know I was doing project management or that there was such a job but I was starting to run projects. I started doing sales support for market data services and grew into managing development projects.

I went on to work at Merrill Lynch, Chubb & Son and Mitsui Sumitomo Insurance, gaining responsibility for more complex projects, mostly in IT. I discovered and joined the Project Management Institute (PMI) in 2000 (while working at Merrill Lynch). That was a huge step for me. I quickly got my Project Management Professional (PMP) qualification and starting speaking and writing.

At Mitsui Sumitomo I began working closely with the top senior managers: doing strategic planning and helping them execute their strategic plans through projects. Recently I have gotten involved in other professional organizations, including the International Project Management Association (IPMA), the American Society for the Advancement of Project Management (asapm) and the Association for the Advancement of Cost Engineering (AACEi).

In June of 2007 I decided to work full time for my own company: Real-Life Projects. I wanted to be able to share

what I had learned with as many different people as possible wherever those ideas were going to have an impact.

**Q: Starting out on your own must have been a big step. Do you feel it was worthwhile? What were the highs and lows?**

It was definitely worthwhile. I enjoy the life and the freedom of running my own company. Having previously set strategy and goals for projects it is a joy to have the authority to set goals and strategy for my own company now. Creating my own business plans has been a real high for me.

Occasionally there are down days. Marketing is still hard for me and sometimes I get caught up in problems with accounting or computer issues which can be frustrating. I have professionals who I can call on for help but I am ultimately responsible for everything. It is so frustrating to plan out a day for follow-up calls only to get sidetracked on some administrative issue. I set tough goals for myself and I get frustrated when I cannot meet them.

**Q: What are you currently working on?**

The work I do for clients is usually six-weeks or less in duration. In between I do marketing. My schedule includes some dates teaching classes in project management and scheduling. I have proposals out for review to help improve scheduling processes and I recently completed some recommendations for an Azerbaijani insurance company's automation.

**Q: Through your work you must meet a wide range of people with differing needs. What do you perceive as the greatest current challenge for project managers and how do you help your clients overcome it?**

The greatest challenge is always communication. People blame project failures on missed dates, budget problems, technology or other issues. But ultimately a project failure is a failure to communicate. If we communicate well then schedules and budgets should be revised. If we communicate well then we should be able to overcome technology problems and disappointments.

Personally I have spent most of my career working to become a better communicator. I am an introvert but I joined Toastmasters to learn public speaking and I now speak

“Ultimately a project failure is a failure to communicate.”

several times a month. I practice all the time. I often recommend Toastmasters to project managers I work with as a way to do life-long learning in speaking, communication and leadership.

**Q: *Joining Toastmasters looks like a great way to improve your communication skills. What would you say are the top things that being a member has taught you?***

The biggest and most important lesson has been that mastering any kind of communication is a lifelong journey and that journey is all about practice. I used to think some people were just born to be better speakers or writers. Now I know that the best speakers and writers practice all the time. They share their best material when they perform but they practice, practice, practice. Toastmasters has given me the chance to speak as often as I want and try new things in a receptive, supportive and no-risk environment.

I have learned more specific techniques that I can count including being able to tell a joke in a serious speech, opening a speech with a gripping start and how to call a meeting to order when there are dozens (or even hundreds) of people in the audience.

Toastmasters is a great programme and I really do highly recommend it.

**Q: *What do you think has been the most important change in project management over the last 10 years?***

I still consider myself a “youngster” in this field and I do not know what to pinpoint. The two most critical things for me in the past 10 years have been:

- Discovering the community of project managers through PMI, IPMA, asapm and AACEi
- Starting to link organizational strategy to project execution

**Q: *What is your number one project management tip, trick or technique?***

Project charters. We all have them but many of us do not understand them. Understand who authorized you to run this project and what the project is supposed to accomplish. All project success starts from the charter.

**Q: *What characteristics do you think it's important to look for when bringing people on to a team?***

An ability to work together. No one can be successful on every team. The people have to fit together and learn to work together. If you ever have a great team working together then you should try not to break them up.

**Q: *How much domain and/or technical knowledge do you think someone managing a project has to have?***

To get the work done people need to understand the work of the project. Occasionally I have seen some project managers succeed without domain knowledge but only with a trusted adviser who is a domain expert. I prefer a domain expert when selecting a PM.

**Q: *What is your biggest bug bear/gripe/pet peeve?***

Currently it is the PMI standards process. I think the Project Management Body of Knowledge (PMBOK) Guide is a great document but the recent versions suffer from a disconnect from reality. There is a term “scheduling model” that has crept into the PMBOK Guide and it is now spreading to the Scheduling Standard and other PMI standards. It is a word I have never heard a real scheduler

“Mastering any kind of communication is a lifelong journey”

use. I just got an e-mail from someone saying that the 4th Edition of the PMBOK Guide has brought back the technique of PERT estimating which has been entirely replaced by Monte Carlo methods among professional risk managers. My own attempts to protest to the term “scheduling model” in an exposure draft lead to no changes and I worry that PERT will now be put back in the PMBOK Guide. I fear that the new standard teams are out of touch with the real practice of project management.

**Q: *I find it to be a common attribute of heavyweight standards and processes that they can quickly lose their real life applicability. Why do you think this happens?***

Personally I like standards that are short, specific and customized. There is a place for long and complex standards, like the ones published by PMI, but for real customers I try to boil down standards and guidelines into a few pages.

Complex and long standards become so difficult to understand that you need an expert in the standard itself in order to use it and modify it. The more time the experts spend with the standard itself, the less time they spend actually applying it and actually running projects. The standards then become more out-of-touch with real life. As they become more out-of-touch they also usually grow and become more complex. This cycle repeats thus making the standard become less and less useful over time. Eventually someone may produce a new standard or they might simplify the standard to make it more practical again.

***Can you tell us a bit more about your experience of PERT and Monte Carlo methods and why you think Monte Carlo based approaches should be preferred?***

PERT is based on a simplified formula of three-point estimates. The formula describes a normal bell-shaped distribution curve. There is a likely outcome and the likelihood of a higher or lower value drops off according to a well-known statistical formula. It works well for random and even distributions of risks and outcomes.

Unfortunately project risks are not even, random or normal. Few real-life risks follow a bell-shaped curve. For instance it might be likely that a coder can create a specific module in two weeks. That is our “likely” estimate. It is possible that they can get it done in 1.5 weeks: that is our “minimum” estimate. What is the maximum time? There is really no limit to the maximum possible time it could take. The minimum estimate we can reasonably estimate but not the maximum. For the sake of argument, we will say the maximum is 10 weeks.

PERT would take these three estimates and try to fit them into an even bell-curve distribution. Because the distribution is highly skewed, with a small amount of possible improvement over the likely outcome, but a large amount of possible delay, the bell-shaped curve will never capture the estimate well.

Monte Carlo software lets you select the correct shape of the distribution curve from several different choices. Most people would select a “triangle” distribution with the minimum of 1.5 weeks, a peak at 2 weeks and a steady

roll-off in likelihood going out to 10 weeks. A good software package will offer ten or more different types of shapes of probability curves and then it will mathematically calculate the shape of the curve of the final result (such as the final end-date).

Monte Carlo is much better able to correctly model real-life project risks.

I tried using PERT once but the minimum and maximum estimates were simply not believable. With Monte Carlo it takes some time to build a good model of my risks but the final graphs are more realistic and believable.

“Monte Carlo is much better able to correctly model real-life project risks.”

***Q: How do you see project management evolving into the future?***

It is clearly growing as a recognized tool and discipline. I expect that there will be tremendous discussion around how to integrate project management into the whole enterprise. Project Management Offices were a start but I do not think they are right for every organization and they may not work well in the long-term. Programme management and project portfolio management will become increasingly important topics in the next ten years.

# Interview: Johanna Rothman

5<sup>th</sup> March 2008

Johanna Rothman runs her own project management consultancy firm Rothman Consulting Group, has written a number of influential project management books, has had numerous articles published and blogs regularly. This week Johanna kindly managed to squeeze me in to her hectic schedule and gave me her views on the problems facing project managers and the best ways to solve them.

**Q: Many people may already know you as an author of several project management books or from your blogs on *Managing Product Development and Hiring Technical People*. Can you give us a brief summary of your career to date?**

I started as a developer in 1977. I managed small projects for a few years as a technical lead and then became a tester for a couple of years (starting in 1985). I started managing larger projects in 1986 and, in 1988, I

“A serial lifecycle (waterfall or phase-gate) is the hardest lifecycle to try to predict anything about.”

managed my first programme and became a manager of Software Quality Assurance (SQA). In 1990 I became the head of a small business unit. That was followed by a few other middle management jobs in development and SQA and finally starting my consulting business in 1994.

**Q: What are you currently working on?**

I have several writing projects: my next book about portfolio management, a chapter for the Beautiful Teams book and other articles. I'm working on a new estimation workshop. I have client work that I can only talk about in general terms: a couple of assessments and project management training.

**Q: In your work as a consultant what do you perceive as the greatest current challenge for IT project managers and how do you help your clients overcome it?**

Senior managers want to see results out of IT. But most senior IT managers (CIOs and their counterparts) don't know enough about how to measure projects or about how to think about project success. So they ask their Project Management Offices (PMOs) to mandate a single process for all projects and therefore hamstringing their project managers. The second greatest challenge is that the project managers have the same thinking.

When I come in, either to do an assessment or to do some project management training, I can help the project managers see that they have alternatives and that they need to assess their project's context to choose an alternative. With any luck I get to talk to the senior managers about their need for process. Their perceived need for process is really a surrogate for not knowing what to measure.

**Q: One of the areas you specialise in is product development. What specific project management challenges do you think are associated with product development?**

One of my clients is struggling with this now. How does a project manager help a product manager create a roadmap and how do they help a senior manager (or PMO) manage the portfolio so that the project manager just has the necessary features to manage in a project?

Each product typically has several releases and an organisation has to balance which releases of which project they want when. A project manager has to be able to provide a ballpark about either how long a release will take and/or how much a team can put into a release to make sure the project manager is delivering the value the organisation wants.

By the way a serial lifecycle (waterfall or phase-gate) is the hardest lifecycle to try to predict anything about.

**Q: That's interesting. I think that many people would assume that a serial lifecycle gives you the greatest predictability. What leads you to hold the opposite viewpoint?**

With a serial lifecycle you don't have any real knowledge about the system until integration and test. You have no meaningful data until integration and test. All the data you

have is surrogate data. So you have the least amount of predictability until the end of the project.

With an incremental lifecycle you can take the prediction you made about the first chunk you did and learn from it. If you thought that chunk was going to take 1 week and it took 2 you can consider what that means for your project. In an iterative lifecycle you can learn from the prototypes you did to see if things will work out the same way as you thought. In an agile lifecycle you know what you thought you could do in the first timebox and use that for the next timebox.

But with a serial lifecycle you have no predictive powers until you reach final integration because that's when you can finally measure what you've been doing. And, if you leave all the testing until *after* integration (which is not what you're supposed to do but is what many people do), you really have no predictive power until after you do one round of testing. And many projects use only manual testing so it isn't until they're 3 or 4 weeks into testing that they realise they have *months* of work ahead of them rather than weeks.

**Q: What do you think has been the most important change in project management over the last 10 years?**

In the last ten years I've seen a huge emphasis on certifications for project managers. It's important because it's a red herring.

Gaining a Project Management Professional (PMP) qualification is a laudable goal. Anyone can learn something while studying for a PMP. But becoming a PMP is no guarantee of being able to manage a project. And the change I've seen is that companies are looking for PMPs but not helping their staff learn how to manage projects successfully.

**Q: What is your number one software project management tip, trick or technique?**

If I have to pick just one it would be rolling wave scheduling. The way I do it is this:

Lay out the major milestones.

Schedule the next 2-4 weeks (no more than 4 weeks) in detail with inch-pebbles.

As you finish one week, add another to the end of the schedule.

If you only schedule for 2-4 weeks then you can build in adaptability to what is actually going on with the project. If you can adapt to reality then you're in great shape.

**What characteristics do you think it's important to look for when bringing people on to a team?**

I look for people who can mesh with the already-existing team. A team that knows how to work together will figure out what they need to do to be successful. That means that I look for solution-space domain expertise and functional technical skills but, even more, I look at the culture of the team and hire for that.

**How much domain and/or technical knowledge do you think someone managing a project has to have?**

A project manager has to know enough about how the people work and what they're working on to assess risk. In *Manage It! Your Guide to Modern, Pragmatic Project Management*, I wrote a sidebar "How Technical Does a Project Manager Need to Be?" that addresses this question in more detail.

**What is your biggest bug bear/gripe/pet peeve?**

Schedule games, especially the Split Focus (multitasking) schedule game and the Pants on Fire (we-can't-decide-which-project-is-most-important) schedule game. Split Focus wastes time for everyone and prevents projects from moving forward. Pants on Fire doesn't allow a team to finish a project. Why don't senior managers realise how damaging either of these games are for the team?

**Q: That is a very good question. Do you have any idea what the answer might be?**

Too many senior managers don't manage the project portfolio: the choice of which projects to start and stop when. To be fair I don't think they know that they have to or know how. Because they don't, and because senior managers multitask in order to do their work, they forget (or don't know) that multitasking is dangerous for technical staff.

“Too many senior managers don't manage the project portfolio.”

**How do you see project management evolving into the future?**

Project managers will have to become comfortable with Agile approaches to the project and the practices. That means that project managers will become more adaptable. I hope that senior managers will start evaluating the project portfolio to avoid the schedule games.

# Interview: Raven Young

9<sup>th</sup> June 2008

Seattle based Raven Young is a freelance IT project manager and writer of the excellent PM blog Raven's Brain ([www.ravensbrain.com](http://www.ravensbrain.com)). This week I had the pleasure of catching up with her to find out a little more about her views on the past, present and future of IT project management.

**Q: Most people will know you from your excellent project management blog Raven's Brain. Can you give us a brief summary of your career to date?**

I got into "management" in my early 20s. I was working in retail back in the early 90s and was about the only person that knew how to use a computer beyond the store's Unix based inventory application. I quickly moved into a management position and worked my way up to the impressive sounding title of West Coast Regional Operations Manager at the age of 23. My desire to plan and reduce costs and rework carried me through this phase of my career where I worked in retail store set-up, new construction and operations management for Crown

“Communication is about more than speaking and writing.”

Books, and eventually my role stretched to encompass technical projects. This is where I found the joy of project management in the high tech industry and I've now been working on web and software development projects for the last 14+ years.

**Q: What are you currently working on?**

Over the past three years I have been fortunate enough to be self-employed, which has afforded me the benefit of working on a number of different opportunities, each offering their own unique experiences. I've continued to take on short-term assignments, acting as a business, management and leadership coach and working on different project solutions. I just wrapped up a brief stint contracting as a Programme Manager III for Microsoft which was an interesting experience in itself. I also continue to build up my Project Management blog Raven's Brain focusing on the "soft-skills" that are so essential for Project/Programme Management but are totally absent from the PMBOK and other Project Management texts (and essentially missing in many PMs).

**Q: What do you see as the greatest current challenge for project managers and what do you personally do to overcome it?**

Communicating *effectively*. We send emails, conduct & facilitate meetings, distribute reports, metrics and schedules, send memos and notes, have chats in the hallway and ad-hoc meetings, but I believe PMs still aren't doing enough to communicate the *right* information at the *right* time to the proper audience and ensuring the message conveyed is both heard *and* understood. Sometimes we overload folks with information or bury key points or action items in long winded emails or speeches. Other times we are so sure of our message we don't think that others might not "get" it or understand what is expected of/from them.

I try to publish action items the same day as they are discussed and make sure everyone understands their role and what is expected of them. I also strive to be brief in emails and write clear and concise messages, highlighting any actions required. Most importantly, I try to remember that communication is about more than speaking and writing. It also involves listening, understanding, confirming and sometimes restating, interpreting and adjusting your message to your audience and so much more!

**Q: What do you think has been the most important change in project management over the last 10 years?**

The interest in project management as a profession and the desire for PM memberships and certifications is at an all time high. When I joined PMI (Project Management Institute) in 2001 they were boasting global membership numbers of 30,000. Now PMI has over 260,000 members in 171 countries and similar project management organisations are also showing heavy growth. I believe that joining an organization focused on furthering your career is a positive and you can't deny the fact that most project management jobs are requiring some type of certification. However I do fear that too much stress is placed on possessing a certificate rather than one's actual abilities to get the job done. I am a member of PMI but do not possess a PMP cert and, though I don't doubt much is learned in achieving the PMP cert, I would encourage potential employers to look beyond certification to ensure

you are getting a “whole PM”, with all the necessary skills and experience to do the job.

**Q: What is your number one PM tip, trick or technique?**

When creating a schedule/plan don't forget to account for non-project hours. There are basic things to account for like vacation, sick and holiday but don't forget training, off-sites, company events and the more unpredictable time-eaters like jury duty, death in the family,

“When creating a schedule/plan don't forget to account for non-project hours.”

maternity/paternity and medical leave, accidents, etc. You can even factor in team member productivity (new employee ramp up time, senior dev coaching time, all day meetings and other commitments) and adjust hours worked each day from 8 to 6 (to account for answering emails, hallway conversations, meetings, etc.) if you want a more accurate picture of hours available. I'll be the first to say that a lot of execs don't want to see this picture (in some cases your available project hours will get cut in half when factoring all this in) but it is the best way to identify the amount of hours your team has available for working on a project.

**Q: What characteristics do you look for when bringing people on to your team?**

If you already have a team in place that you are adding to then the most important quality to look for is someone who will be a good fit with respect to team dynamics. You want an individual who will enhance the current team's skills and make everyone better, not a disruptor. Qualities like drive, initiative, good communication and collaboration, flexibility and adaptability are all important but inconsequential in the end if the team doesn't gel.

**Q: How much domain and/or technical knowledge do you think someone managing a project has to have?**

The best answer I can give is really a non-answer: enough to get the job done. If you are working on a highly technical effort and the expectation is such that the project manager should be able to architect or dissect the system and trouble shoot issues then you would want someone with a technical and/or development background who is comfortable in that role. If you have directors of development, tech leads, a system architect or other technical resources on

your project then you probably don't need a PM that is able to sit down and code a particular piece or help find errors in code reviews.

My thought has always been that a PM is there to focus on driving the project and, though technical skills are always a positive, someone who understands the basics, has managed other technology projects, knows how to ask good questions and is willing to learn how the system works (front end, back end, data model, jobs, reporting, APIs, etc.) can be just as successful. In fact, they can be more successful than a PM with a strong technical background if you agree with the argument that they are better able to focus on the overall project without getting mired in technical details.

**Q: What is your biggest bug bear/gripe/pet peeve?**

Organisations that hire project managers but don't buy-in to the process. I can't tell you how many times I've spent hours creating a realistic schedule that the project team believes in, only to have upper management say “that just won't work - we need it by X”. This causes everyone to make up their project estimates to match the desired end date and it creates a culture where nobody believes in the schedule (I wonder why??). When you start to hear team members say things like “just tell me when it's due and I'll give you my estimates” you know there's a problem. This can be a difficult situation for project managers and it's easy to give in to the VPs and “make it happen”. That short term thinking will only burnout your team and drive you, as the project manager, insane.

My advice is to stand your ground, explain and show how you created the schedule, offer trade-offs in scope or ask for more time and continue to state your concerns and document them. If the project fails to meet the desired (required? forced?) deadline you can discuss the schedule issues at the post-mortem and suggest that the next project follow tighter project management processes.

**Q: How do you see project management evolving into the future?**

Project Portfolio Management (PPM) will continue to grow in use over the next few years and will become a larger focus of upper management and executive teams as they look to increase profits, start a new venture or simply

“A project manager is there to focus on driving the project.”

survive in the current tenuous economy.

There is already a growing interest in PPM and more organisations are going to have to learn how to align strategic goals and project management through PPM processes going forward. Eric D. Brown, talking about Information Technology Leadership & Alignment, said this about IT project alignment recently: “These projects must fit into your strategic goals and must deliver value to your organization.” That’s exactly what PPM is all about: working on high value projects, as defined by the organizations goals, to get the best return and ensure the

PM and team are working on only approved, value-aligned projects. Besides, if you don’t prioritize the projects in your portfolio, how can you know what to work on next? Which brings the most revenue or provides the bigger impact to the customer?

PPM is here to stay and, if you work on multiple projects or in a multi-project environment, you should read up on the subject. Johanna Rothman’s blog is a great place to start. She also has a forthcoming PPM book I’m looking forward to. It’s still in the early writing stages but I’m already waiting for her great thoughts on all things PPM.

“Project Portfolio Management is here to stay and, if you work on multiple projects or in a multi-project environment, you should read up on the subject.”

# Interview: Jim McCarthy

15<sup>th</sup> June 2008

Back in 1999 I read a great book about software engineering: Dynamics of Software Development by Jim McCarthy. This book was one of the first to express the idea that software development is primarily a people-centric activity and that therefore the success of any software project is intrinsically bound to how well the team works together.

Jim has had a wide ranging and successful career in the software industry and I was delighted to have the chance to interview him and discover his current views on the management of software projects.

**Q: Many people will already know you from your books *Dynamics of Software Development and Software for Your Head* or through your *BootCamp* training course. Can you give us a brief summary of your career to date?**

I began my career as a software guy over 30 years ago. Over the years I have synthesized what I learned from software development and corporate experiences and applied it to solving the riddles of team dynamics. I have had responsibilities in development, testing, marketing, programme management, user education and general

“I have synthesized what I learned from software development and corporate experiences and applied it to solving the riddles of team dynamics.”

management. I led software development teams at Bell Labs, The Whitewater Group and Microsoft Corporation. Since 1996 I have devoted myself to researching groups and how they create products and organisations. We use a teamwork lab (McCarthy BootCamp, a product development simulation) and in situ research at corporations, large and small, worldwide.

I wrote Dynamics of Software Development in 1995 and Microsoft has recently released the 2006 edition of Dynamics which is cool because it includes both the Core (see below) and the original video on which the book was based. My wife, Michele, and I have created the Core Protocols (now available in version 3.0) which are a collection of interpersonal protocols that support results-oriented behaviour, the efficient aggregation of individual qualities into a greater whole and the development and realisation of shared visions. Together, we authored Software for Your Head and now host a podcast show on team and other business issues. I also give numerous speeches and seminars. I can be reached at jim@mccarthy.net or via our website www.mccarthyshow.com.

**Q: What are you currently working on?**

Once there was this guy interviewing me in Malaysia and he asked me “What software are you working on now?” I answered, honestly, “Yours.” In a way I guess that answer still holds but in this case for you and your readership. Specifically, however, I am working on the Core V4.0, doing speeches and BootCamps and coaching teams and individuals.

**Q: Can you tell us a little more about *The Core and BootCamp*?**

The Core Commitments and The Core Protocols are a collection of codified best practices that came out of our teamwork laboratory (called BootCamp). They have been compiled over an eleven year period and are now in Version 3.1. The current version can be downloaded from www.mccarthyshow.com.

We didn't create The Core.

Instead we watched it grow. We did create the set of initial conditions under which The

Core protocols, or something very much like them, would almost certainly emerge. Over the years we have maintained healthy conditions for Core evolution.

A proper credit also has to include the thousands of BootCamp students from around the world who contributed to The Core's development over the years.

**Q: Through your work you must meet a wide range of people. What do you perceive as the greatest current challenge for software development managers and how do you help them overcome it?**

The greatest current (and past and future) challenge for software development managers, and for all humans everywhere I suspect, is accurately perceiving reality and effectively accounting for it in their behaviour. Some people call this integrity but that's kind of an overloaded word. It is, however, an accurate term if you mean it as you mean it when you discuss the "integrity of a structure". Much of the time, either our perceptions depart from reality or, if they don't, our behaviour does not have the integrity of our perceptions. In either case we are misbehaving in the most fundamental sense.

**Q: Dynamics of Software Development was first published over 10 years ago now. Since then what do you think has been the most important change in software project management?**

That's kind of a trick question from my point of view. The obvious answer would be to say Agile. Or XP. While I give Agile and XP practitioners and developers and authors all due respect I honestly don't think that originality is the chief virtue in Agile or XP. There has been little that I'm aware of that goes beyond what was covered in Dynamics of Software Development and much of that book was a build on the work of many previous others. Agile especially has done a remarkable job of popularising basic ideas. This is no small trick. However, on a good day at least, I like to believe that we have been conducting much more original, fundamental and even transformational work than is currently popular.

**Q: The original book contained 54 "rules" for software development. Which one do you feel still holds the greatest relevance today and why?**

"Team = product". All the virtues and vices of the team express in the product, and vice-versa.

Or possibly "Establish a shared vision". The relevance of shared vision is greatly intensified now that we have discovered how to do it with any project and on any team.

**Q: Can you tell us a little about the ideas behind your latest book Software for Your Head?**

If you picture a kind of TCP/IP for humans. Or imagine a suite of interpersonal protocols that can be used to help any team get to a state of shared vision and achieve the highest possible level of collaborative intimacy then you will glimpse the vision behind Software for Your Head.

**Q: What is your number one software project management tip, trick or technique?**

Discussion should be illegal. Less talk, more code.

**Q: What characteristics do you think it's important to look for when bringing people on to a team?**

Make sure the team itself is well and truly "booted" (i.e. aligned and in a state of shared vision) and let them handle the problem however they see fit. Generally, although the problem feels like too few people are on a project, it is usually the case that too little of the many people assigned to the project is actually engaged.

**Q: How much domain and/or technical knowledge do you think someone managing a project has to have?**

Not as much as all that. Generally, a person who understands the many ways people trick themselves out of success is much more valuable than a person who has

“Imagine a suite of interpersonal protocols that can be used to help any team get to a state of shared vision.”

particular domain or technical expertise. Obviously, having both types of knowledge is probably better than having one.

**Q: What is your biggest bug bear/gripe/pet peeve?**

When I am certain something is absolutely true it is almost certainly false. That bugs me. It's a sort of Uncertainty Principle of its own kind. It is annoying that somehow my personal degree of certainty about the truth or falsity of something seems to inversely (and disproportionately) affect its actual truth or falsity.

# Jim McCarthy on The Core

In my interview with Jim McCarthy I asked him the following question:

*Q: Can you tell us a little more about The Core and BootCamp?*

It may be a short question but to do it justice Jim McCarthy sent me a full and detailed response that mirrors closely what is published on The McCarthy Show website ([www.mccarthyshow.com](http://www.mccarthyshow.com)). I found it fascinating and, although I felt it necessary to edit it significantly for the interview write-up, I am pleased to present it in full here:

The Core Commitments and The Core Protocols are a collection of codified best practices that came out of our teamwork laboratory, called BootCamp, compiled over an eleven year period, and now in Version 3.1. The current version can be downloaded from [www.mccarthyshow.com](http://www.mccarthyshow.com). The story of the evolution of these interpersonal protocols is probably worth telling in some detail because a) they actually work, and this is because of their mode of genesis and b) it makes a good story.

We didn't create The Core.

Instead, we watched it grow. We did create the set of initial conditions under which The Core protocols, or something very much like them, would almost certainly emerge. Over the years, we have maintained healthy conditions for Core evolution. Along the way we also pruned the tree from growing into a few false directions. And we added resources: our own money, time, focus, and stamina. We protected it. Took notes. Tried it out. Passed it out.

A proper credit also has to include the thousands of BootCamp students from around the world who contributed to The Core's development over the years.

Crediting one person or segment of contributors exclusively would be inaccurate, however. The real story is both simpler and more complex.

The emergence of The Core was in some measure a result of our experiences in 1994-96. We were working for a commercial software company, leading a development team of approximately 150 people. We used a homegrown aphorism to help us try new ideas:

Team = Product

That's the idea. Because of its many virtues, despite its deficits, and regardless of others who have had the same thought, this maxim became a bit of a mantra for us. During stressful times, when we were tempted to retreat from the overwhelming complexity of the project tasks; when the confusion and disorientation were really getting to us; when schedules were slipping and goals receding and prospects were looking pretty grim indeed. Then, just when we needed it most, someone in our group would invariably come up with a new idea, would provide a fresh point of view based on "Team = Product." "I get it," he might say, and then rattle off some new application of "Team = Product" that could apply to our situation. Occasionally, these ideas were profound; more often they weren't. They were almost always useful, however.

The essence of the "Team = Product" philosophy is that the behaviour of a team maps directly to the qualities of its product, and vice versa. If you want a product with certain characteristics, you must ensure that the team has those characteristics before the product's development. We also realised that everyone has a product or provides a service. Everyone produces a concrete expression of his value system that carries that person's virtues and vices out into the world.

What was our leadership team making? We moved through the hierarchical levels in our organization and

“Everyone produces a concrete expression of his value system that carries that person’s virtues and vices out into the world.”

answered two pertinent questions at each interesting point: Who is the team here? And what is its product?

Let's call the team of frontline developers the Level I team. Level I makes the actual product. The managers of this team constitute the Level II team. Its product is the Level I team. When applying the "Team = Product" philosophy, the team on one level is the product of the team at the next higher level. If the Level II team sees an undesirable trait in the Level I team, it must be an expression of or reflection of Level II teamwork and the Level II team members. This pattern applies to teams at all levels, right up through the corporate ranks.

This idea may seem clever, obvious, fanciful, or just plain wrong-headed, but to us it was certainly helpful. Using this model, no one can hide from accountability. In our situation, even though we were bosses, we could not fault a team for lacking a virtue, unless and until we had personally demonstrated it. Nor could we expect any remedy that we weren't personally modelling. On the one hand, this realisation was depressing, because there really was no escape: responsibility inevitably migrated upward and weighed heavily from time to time on our well-paid, if under-exercised, shoulders.

On the other hand, this realisation offered an incredibly hopeful perspective as something more, something immediate, something completely within our control that was available to remedy any shortcomings of the team. If we saw something screwed up somewhere or noticed some good fruit dying on the vine, we could immediately find and fix the problem. To inspire other team members to go get that fruit before it died, we would gather and visibly devour tantalizing fruit that had gone unpicked in

“So many new possibilities opened up at such a rapid pace that we were unable to keep up with them.”

our own neck of the woods.

If we wanted any property to materialise on the Level 1 team, we would have to incorporate that property into our own behaviour. This change in behaviour was conceptually simple, but challenging to implement. In any case, keeping this basic framework in mind exposed many novel approaches to team problems. When we first applied this perspective, so many new possibilities opened up at such a rapid pace that we were unable to keep up with them. Although many little tests and a few big ones did yield the desired results, we saw so many new solutions to problems that had plagued us for years that we hardly knew where to begin.

We quickly realized that we couldn't possibly conduct sufficient experiments to develop a full understanding of precisely how useful the formula was; to discover where it failed; or to see where the behaviour it inspired might lead. We wanted to explore its dynamics and map its aetiology in the systems we believed it governed. That is, check it out all the way. Unfortunately, the experimental opportunities in a commercial software development effort are necessarily limited. A major obstacle is the simple passage of calendar time. A large commercial software project can take months or years. The

possibilities we were seeing appeared so valuable, however, that even a few months seemed far too long for each cycle if we were to learn everything possible. With millions of dollars at stake on a single development effort, radical experimentation seemed risky. The number of variables with which we could tinker was low. Together, the sluggishness of “real-world” calendar time and the responsibilities of prudent business practices worked against the idea of implementing the sustained, radical, and rapid experimentation that we envisioned. Still, we thought big breakthroughs in team dynamics were possible-breakthroughs that could make collaboration simpler and more effective for any team.

To study this material in depth, we had to complete a development cycle in a much shorter time. Life itself was too short to go through enough development cycles. Even a very busy, unusually stable, and highly focused development manager could-if she stayed with the task for a long time-expect to oversee 10 to 20 projects in one professional lifetime. Many of these projects would use essentially the same teams, reducing the diversity of team sources that would enrich the manager's education and hasten experimental progress.

In early 1996, to accelerate the rate and breadth of our experiments, we went out on our own and established a laboratory devoted to the study and teaching of teamwork. The ultimate existence of The Core protocols became a virtual certainty when we decided how we would operate the new lab, which we named “BootCamp.” The principal experiment conducted would be a recurrent product development simulation, lasting five days and nights with a new team each time. It would take place every month or so. The team members would complete four steps:

- Form a team.
- Envision a product.
- Agree on how it would be made.
- Design and build it.

At the end of the week, the teams would have to deliver their products on time, or stay longer to do so, or not, as they chose. We knew that we could successfully conduct such a product development effort, even leading it personally, if needed. We had done just that for many years, earning our living in a variety of environments. We had sufficient information, tips, techniques, and useful practices to transmit high value to most students. We could teach them practices that could ensure the successful outcome of their own product development efforts, now or later, simulated or not.

We had already gained, organised, and articulated considerable knowledge from our experiences in leading or otherwise contributing to dozens of project efforts, most of which proved quite successful. This body of

knowledge would serve as the starting point for the first BootCamp teams. Even if we learned nothing during BootCamp, we still would have plenty to offer. BootCamp has allowed us to effectively compress a project development cycle into a five-day experience. In five days, students learn what would normally require a long development cycle. The intense BootCamp experience includes all of the failures and triumphs that occur with normal team formation; the creation of a team-shared vision; and the design, implementation, and delivery of a product. The days in each BootCamp are packed with accelerated team dynamics; what usually takes a year or more is created in a few long days and nights of exceptionally deep engagement.

The many new insights from BootCamp emerged at a vastly increased clip. The learning pace was accelerated by our experience of working intimately with hundreds of different project teams. We first helped to create the team, and then their products. We experienced complete development cycles with incredible frequency and velocity: 6 times per month at peak periods.

Working with teams of every kind and composition, and working before and after BootCamp, we applied what we learned to our own teamwork. One additional factor led to the creation of The Core protocols, and originated in our standard assignment to the students. Each team would have to build a product in one week. But what product would the BootCamp teams make?

At one level, BootCamp is conceptually simple: we assemble a group of project team members. Sometimes the students are members of a pre-existing team. Sometimes they represent as many backgrounds as possible: corporate executives, entrepreneurs, computer scientists, writers, editors, graphic artists, engineers, managers, programme and project managers, producers, HR, technical support, administrative assistants, consultants, marketing managers and salespeople.

Often, there will be those who don't work in a corporate environment in attendance: nurses, teachers, homebodies and press members. We give each new team-in-waiting a single assignment:

Design, implement, and deliver a course that teaches you everything you need to know to deliver great products and services on time, every time.

This assignment has remained unchanged since the first BootCamp. It seemed to us that it would be useful to look at team dynamics from the real-time point of view of a team actually working in a state of effective teamwork. Teams exhibiting the most desirable teamwork were best able to solve the riddles of such teamwork. The decision to devote the BootCamp teams' efforts to resolving the

issues of bringing teams to the effective state they were enjoying was a productive innovation. Teams in a newly gained high-performance state produce extraordinary results. When they examine the conditions and elements of their own high performance, as it occurs, the quality of insight is substantial.

Almost every BootCamp team has experienced the following flash of insight: if teamwork itself could be made more efficient and direct, then the team members would be able to find the solutions to the big problems that vexed them. This knowledge could then be leveraged to enhance their other endeavours. High-performance teams typically acquire their reputations by accomplishing the specific goals they set for themselves. For example, a great basketball team wins many basketball games. The players are not remembered for their contributions to the art and science of team enhancement, but for putting balls through hoops. Achieving a team's original goal is a task not directly related to explicitly uncovering the dynamics of team formation. In the case of the BootCamp teams, the presenting task became the discovery, refinement, and codification of practices that would always lead to the formation of great teams.

As one BootCamp led to the next, we began capturing the best practices employed by the teams, and we encoded these behaviours to make them easily transmissible. These lessons from the BootCamp experiences gradually evolved into The Core protocols. When a team applies The Core protocols consistently, it will produce superior results.

The booting process stimulated by The Core protocols

“Teams in a newly gained high-performance state produce extraordinary results.”

can be ongoing, yielding more efficient and capable groups. The lesson that the booting process continues in a general way is reinforced vividly when we see every new BootCamp team learn more, do more, and add more to the richness and the reproducibility of the “multipersonal” patterns and protocols that lie at the heart of The Core.

And that's our story-how we watched The Core protocols emerge.

The Core Protocols are described in great detail in the book “Software for your Head” and in podcasts available on [www.mccarthyshow.com](http://www.mccarthyshow.com).