

Context Counts: Position Paper for SEMAT

By Scott W. Ambler
Chief Methodologist/Agile, IBM Rational

I believe that for the SEMAT endeavor to be successful that as a community we need to come to a consensus regarding both our philosophical foundation and our scope. Ivar Jacobson, Bertrand Meyer, and Richard Soley have made good inroads towards addressing these issues [1, 2] but we still have some work ahead of us.

My main points:

1. Practices are contextual, never “best”
2. We must go beyond practices
3. We’re more successful than we think
4. We need to reuse existing resources

1. Practices are Contextual, never “Best”

We must define the context in which practices will be applied, because the context determines the applicability of the practice as well as how it is tailored. An approach which works well for a medium-sized co-located team in a regulatory compliance situation is likely to fare poorly for a small distributed team developing an informational website. Furthermore context is particularly important for the research behind the practices, because without a clear indication as to the context in which the practice was evaluated it will be very difficult for practitioners to identify which strategies are best suited for them. At IBM Rational we’ve been applying the 1+8 scaling factors of the Agile Scaling Model (ASM) [3] to help communicate the context faced by project teams. A tenth factor, paradigm, is implied. These factors are:

1. *Life cycle scope.* Is your focus on the construction life cycle? On the delivery life cycle? On the full system life cycle (include project identification activities, production, and retirement)? On the enterprise IT life cycle?
2. *Team size.* The strategy followed by a team of 7 people will be different than a team of 25, than a team of 50, than a team of 200, and so on.
3. *Geographical distribution.* A co-located team will work differently than a team distributed across several cubes on the same floor, which in turn works differently than a team distributed across different locations within the same city, which works differently than an internationally distributed team.
4. *Regulatory compliance.* A team which needs to conform to the FDA CFR 21 regulations will work differently than a team which doesn’t need to do so. A team working in an ISO 9000 compliant organization will work differently than a team working in another organization.
5. *Domain complexity.* A team addressing a very straightforward problem, such as developing a data entry application or an informational Web site, will work differently than a team building an air traffic control system.

6. *Organizational distribution.* A team made up of people working for the same division of an organization works differently than a team made up of people from different divisions which in turn works differently than teams made up of people from several divisions. Teams with on-site contractors work differently than teams where some of the work is outsourced to an external organization.
7. *Technical complexity.* Teams building new systems from scratch will work differently than those working with legacy systems and data. Teams working with a single technology platform will work different than those working with several platforms. Teams building only software will work differently than systems engineering teams building both software and hardware.
8. *Organizational complexity.* Teams working in an organization with a flexible culture will work differently than teams working in one with a rigid culture. Teams working in organizations with homogenous cultures will work different than teams in heterogeneous cultures. Teams working in organizations with cultures that are pro-IT will work differently than teams in organizations where this isn't the case.
9. *Enterprise discipline.* Teams working in organizations with effective enterprise disciplined (enterprise architecture, portfolio management, governance, asset management, administration, ...) will work differently in organization where this isn't the case.
10. *Paradigm.* Teams following an agile paradigm will work differently than those following a traditional/serial paradigm which will work differently than those following an ad-hoc paradigm.

These factors affect how you address kernel elements. For example, I'll be so bold as to assert that there will be some sort of requirements elicitation element in the kernel. I recently wrote about how to scale agile requirements strategies by working through the ASM scaling factors and describing how each factor affects your approach [4]. The various factors affected the timing of elicitation, the specification strategy, the tooling strategy, the collaboration strategy, and other aspects of requirements work. This is just one example, but the same holds true for architecture, quality, development, management activities, and other potential kernel elements. The SEMAT kernel needs to recognize the contextual factors faced by organizations and of individual development teams.

2. We Must Go Beyond Practices

Although there's been a lot of discussion around practices we've also recognized that there is a lot more to it than that. At IBM Rational we've found that to be successful you must address the "5 Ps" of IT [5]:

1. *People.* People and the way they work together have a greater effect on the outcomes of a project than the processes they're following or the products (tools and technologies) that they're using. People issues include having visible executive sponsorship, building an environment of trust, empowering staff, focusing on leadership as well as management, recognizing that the primary

- gating factor when improving processes is people's ability to absorb change, and promoting a cross-discipline strategy at both the team and individual levels.
2. *Principles.* We've found both internally within IBM as well as with many of our customers that there is a need to define a common set of principles to provide a consistent foundation to enable effective teamwork and continuous process improvement. These principles help to guide people's decisions when their processes and practices don't directly address the situation which they find themselves in.
 3. *Practices.* A practice is a self-contained, deployable component of a process.
 4. *Products.* This includes the technologies – such as databases, application servers, networks, and client platforms – and tools such as integrated development environments, testing tools, and project planning tools used to create solutions for stakeholders.
 5. *Processes.* The previous 4Ps do not exist in a vacuum, we need some sort of glue to help piece all of this together. Minimally this glue is a lifecycle although more often than not it is a full process or method.

I believe that these five issues must be addressed by the scope definition of the SEMAT initiative.

3. We're More Successful Than We Think

My experience, backed up by recent surveys [6], shows that the way that organizations define project success vary based on their context, and as a result I'm not convinced that our track record is as bad as we think it is. My surveys which ask people how their organization actually defines success reports higher success rates than studies which enforce their own definition of success on respondents. I am convinced that a definition of on time, on budget, and on scope compared against up-front promises is clearly not appropriate for most project teams. Perhaps one thing that the SEMAT initiative can do is start educating people on the futility of this strategy. If the people applying the SEMAT kernel cannot reasonably measure success, how can SEMAT in turn ever be seen as successful itself?

4. We Need to Reuse Existing Resources

There is a significant amount of practice and process-oriented intellectual property (IP) available via open source and similar licensing strategies. For example, the Eclipse Process Framework (EPF) at www.eclipse.org/epf/ includes both process tooling and IP which we could choose to leverage free of charge. This IP includes descriptions of Extreme Programming (XP), Scrum, the Open Unified Process (OpenUP), various agile practices, and other material. Of course there are many more examples of process/practice IP available under Creative Commons licenses. How can SEMAT be successful if we strive to reinvent the wheel?

5. References

1. Jacobson, I. Meyer, B. and Soley, R. (2009). *The SEMAT Charter – A Proposal*. <http://sematblog.wordpress.com/2009/12/19/the-semat-charter-a-proposal/>
2. Jacobson, I. Meyer, B. and Soley, R. (2009). *What is the scope of the kernel?* <http://sematblog.wordpress.com/2009/12/18/what-is-the-scope-of-the-kernel/>
3. Ambler, S.W. (2009). *The Agile Scaling Model (ASM): Adapting Agile Methods for Complex Environments*. <ftp://ftp.software.ibm.com/common/ssi/sa/wh/n/raw14204usen/RAW14204USEN.PDF>
4. Ambler, S.W. (2009). *Agile Requirements at Scale*. https://www.ibm.com/developerworks/mydeveloperworks/blogs/ambler/entry/agile_requirements_at_scale
5. Ambler, S.W. (2010). *Scaling Agile: An Executive Guide*. To be published Feb 1 at www.ibm.com.
6. Ambler, S.W. (2006). *Surveys Exploring the Current State of Information Technology Practices*. <http://www.ambysoft.com/surveys/>