Agile Maturity Model

Ravindar Gujral  
(MNG Technologies)  
Shaun (Shanmuganand) Jayaraj  
(ThoughtWorks)
Agenda

- What is AMM
- Aspects of AMM
- Process Deep Dive
- AMM anti-patterns
- Questions & Discussion
AMM- Agile Maturity Model

Let me show you what it looks like.

- It is a mechanism to identify how mature are the agile practices that the team is adopting.
- It is process agnostic (scrum, xp).
- It is a health check for the team and the organization.
- It is a quick look up to help you think through adoption of agile practices.
- It is a discussion starter or maybe a fight starter depending on your team.
Whats AMM & more importantly what AMM is not

- A framework to help organizations be more agile
- Aspects of agility
  - Testing
  - Source Control Mechanism
  - Collective Code Ownership
  - Collaboration
  - Responsiveness to Business
  - Assurance and Governance
  - Story Formation
  - Simplicity
  - Build
- AMM is NOT a mechanism to measure performance of a team
- Don't confuse this with CMM.
Testing

- Two simplified aspects:
  - Functional testing - Done by trusted QA. Is it a repeated quality check for you?
  - Test Driven Development - It helps you create a Flexible design. Adapt to changing and future requirements.

- So where are you? Are these aspects automated.
Collaboration

- The backbone of a good team is its collaborative nature

- Is their infighting when things go wrong?
  - Pay attention to the behaviour of collective responsibility of success or failure of the project.

- Does everyone know when you are failing, more importantly do they know this soon enough?
  - Trending of the success or failure is evident to everyone.
Collective Code Ownership

- Do you have team members that specialize in certain area? What will happen if they get hit by the bus?
  - Are you spreading knowledge across team members. Reduce the "bus factor".

- Do your team members say that they are more productive on their own? Who needs pairing?
  - Show them studies that have shown two developers are more productive working in a problem space.
Build

- Do you check in code and find out next day if everything worked?
- How do you do the quick 1 minute TDD feedback cycle?
  - Fail fast - quick feedback
- Can you cut a CD tomorrow for the customer? If not, why?
  - Provide continuous verification of your product integrity.
- Is your deployment process well understood and mostly automated?
All we can say here is that this will make or break your project.

Have an SCM that supports agile behaviors that you are working so hard to instill (TDD, Continuous integration, etc.).

A good SCM will provide happiness to your most valued resource: Developers.

A good SCM encourages frequent check-ins which in turn saves you from merge hell.
Story Formation

- Do you have a customer creating requirements that can be achieved in your given iteration?
  - ACDC: Define your customer, Let the customer define the deliverable, Let the customer define the acceptance criteria.
- Are your stories true representation of value add to your product?
- Remember Acceptance Criteria != Design
Simplicity

- Are you trying to solve all world problems while developing your flexible design?
- Less said here the simpler it is to understand.
Responsiveness to Business

- Does your customer (euphemism for an MBA manager) make you sign in blood the deliverables at the start of the year?
  - Remind them that business world is always changing (Remind them of the MBA 101 class they took). What will you do if the customer changes his mind?

- Do a value stream map. Reduce cycle time: the time from the requirement to the software back in the customers hand.
Assurance and Governance

- Does your business understand the value of plans?
- Does your business understand that you are not tracking to the plan, plan is tracking reality?
- Do you have Information radiation? Is there a reality check?
- Are you tracking business value with each feature added?
- Is your business doing global portfolio planning?
AMM anti-patterns - Self Targeted Missile

- Using AMM to measure your team's/individuals performance
- Trying to do every practice in the AMM framework
- Trying to game the framework to give yourself better scores
- Using AMM as a manager tool instead of a tool to improve your team and yourself.
- Using AMM to say that you have achieved agility. Following agile practices is about having the right behavior.
Conclusion

- Treat it like a health check or a quick reference.
- Use it as a retrospective tool.
- Use it as a tool to start a discussion.
Q and A