Essential Patterns of Mature Agile Teams

Bob Galen
President & Principal Consultant
RGCG, LLC
bob@rgalen.com
Introduction
Bob Galen

- Somewhere ‘north’ of 30 years experience 😊
- Various lifecycles – Waterfall variants, RUP, Agile, Chaos…
- Various domains – SaaS, Medical, Financial Services, Computer & Storage Systems, eCommerce, and Telecommunications
- Developer first, then Project Management / Leadership, then Testing
- Leveraged ‘pieces’ of Scrum in late 90’s; before ‘agile’ was ‘Agile’
- Agility @ Lucent in 2000 – 2001 using Extreme Programming
- Formally using Scrum since 2000
- Currently independent Agile Coach at RGCG, LLC and Director of Agile Solutions at Zenergy Technologies
- From Cary, North Carolina
- Connect w/ me via LinkedIn and Twitter if you wish…

Bias Disclaimer:
Agile is THE BEST Methodology for Software Development…
However, NOT a Silver Bullet!
The SCRUM Framework

Copyright © 2012 RGCG, LLC

Copyright © 2005, Mountain Goat Software
Outline
Maturity Patterns

1. Truly Emergent Architecture
2. Active Done-Ness
3. Truly Collaborative Work
4. Lean Work Queues
5. Aggressive Refactoring
6. Behaving Like a Team
7. Doing More than Thought Possible
8. Quality on ALL Fronts
9. Congruent Agile Measurement
10. Stopping the Line
11. Investing in Serious CI
12. Pervasive Product Owners
13. Righteous Retrospectives
14. The Power of Complete Transparency
15. Pursue Ruthless KISS
16. Testing is Everyone’s Job
17. The Nuance of a Healthy Backlog
18. Emphasize Strength-Based Teams
19. Performing Extraordinary Facilitation
20. Saying ‘NO’ as a Leader
21. Aligned Organizational Leadership
For each pattern... workshop discussions

- For sets or groups of patterns, we’ll pause and discuss the patterns in small groups

- Looking for examples where you’ve seen the pattern in operation and have a story to tell

OR

- Examples where you’ve seen related anti-patterns in operation and have a counter-story to tell

- Either way, we’ll be looking for group-based discussion around the ways and means of achieving agile maturity
The Patterns
#1) Truly Emergent Architecture

- Comfortable with de-composition on-the-fly; *no BDUF!*
- Drive Sprint #0’s as appropriate
- Backlogs contain learning activity – Research Spike stories
- Should demonstrate architectural evolution in Sprint Reviews
- Architects work in “slices”
  - Perhaps ‘skewed’ a bit forward from other teams
  - Deliver architecture from within the Scrum teams
  - Publish your system metaphors, guidelines, big picture views – to keep everyone focused on goals
#2) Active Done-Ness

- Actively create and automate Acceptance Tests on a Story or a Feature basis
  - Customer heavily involved with definition
  - Not functional tests
- Have established a view to multiple levels of *Done-Ness*
  - Work - Done
  - Story Acceptance
  - Sprint Goals
  - Release Criteria & Goals
- Think in terms of traditional Entry, Exit, and Release criteria
## Levels of Criteria

<table>
<thead>
<tr>
<th>Activity</th>
<th>Criteria</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Team Work Products</td>
<td>Done’ness criteria</td>
<td>Pairing or pair inspections of code prior to check-in; or development, execution and passing of unit tests.</td>
</tr>
<tr>
<td>User Story or Theme Level</td>
<td>Acceptance Tests</td>
<td>Development of FitNesse based acceptance tests with the customer AND their successful execution and passing. Developed toward individual stories and/or themes for sets of stories.</td>
</tr>
<tr>
<td>Sprint or Iteration Level</td>
<td>Done’ness criteria</td>
<td>Defining a Sprint Goal that clarifies the feature development and all external dependencies associated with a sprint.</td>
</tr>
<tr>
<td>Release Level</td>
<td>Release criteria</td>
<td>Defining a broad set of conditions (artifacts, testing activities or coverage levels, results/metrics, collaboration with other groups, meeting compliance levels, etc.) that IF MET would mean the release could occur.</td>
</tr>
</tbody>
</table>
#3) Truly Collaborative Work

- Teams sitting together; work in close proximity
- Avoiding Scrummerfall-like dynamics
  - Stages and gates within the team
  - Long queues with hand-offs
- Members help each other out
- Comfortable pairings (across the team)
- Listening to each other; mutual respect
#4) Lean Work Queues

- Working on as few Stories as possible—driving them towards “Done” before taking on more
  - Fewer things “in process” and small tasks
- Blending roles – individuals doing more themselves and handing off less
- Kanban is interesting variant of the ‘correct’ team behavior
- Think in terms of reducing & eliminating WASTE
- Get it into inventory or production…fast!
Kanban
Iteration-less Production
#5) Aggressive Refactoring

- It’s easy to refactor on new work or greenfield project…so clearly do that.
- But what about hairy, old, fragile code?
- Aggressive refactoring
- Put it on your Backlogs
  - Justify / explain it in business terms
- Remember the relationship to automation – making refactoring effective & Fear-Less
#6) Behaving Like a Team

- Including the Scrum Master and Product Owner
- Developing trust
- Providing each other congruent feedback
  - Getting the “Elephants” on the table
- Spending personal time together
- **Passionate debate; Healthy conflict**
- Strengths & weaknesses; adjust to each; maximizing & minimizing
- Helping one another
- Truly succeeding or failing – as a team; Team-based retrospectives & learning
#7) Doing More than Thought Possible

- Stretch goals within Sprints
- Creative solutions – meeting Sprint Goals and not simply following the Story or Task lists
- Creatively showing alternatives to your Product Owner; getting to “Done” sooner
- The Wisdom of Crowds
- Iterations that lead towards – this is Good Enough
- Fighting *Parkinson’s Law* and *Student Syndrome*
#8) Quality on ALL Fronts

- Leaving behind the notion of “Testing in quality…”
- Professionalism within the team
  - Doing the right things…doing things right
- Self-inspecting; self-policing; self-learning
- Just enough quality, aka, quality has a cost and should be variable based on your context
- Focus on *Craftsmanship*
#9) Congruent Agile Measurement

- Don’t focus too much on metrics; instead on results
- Traditional measures as they can lead to Metrics Dysfunction
  - Measure bugs for reward…get more meaningless bugs
  - Measure LOC for reward…get more meaningless LOC
- Instead measure –
  - **Value** output from the team
  - **Customer** exposed bugs
  - **Quality** practices adhered to (coverage, pairing, refactoring, automation, etc.) and improvements
#10) Stopping the Line!

- Fix your bugs
  - Ruthless testing; immediate testing; immediate feedback
  - Less logging more fixing
- Build is broken; Fix it!
- Need automation for a key area – build it!
- Need to refactor ugly legacy code that is bug infested; Refactor it!
- Key impediments in your team; Resolve them!
#11) Investing in **Serious CI**

- Build on every check-in
- All artifacts – DB code (stored procedures, structure)
- Automated deployments to environments (real and/or virtual)
- **Automation everywhere!**
  - Dashboards
  - Lava lamps
- Serious focus – dedicated team
  - Tools are only part of the answer
  - Develop infrastructure
  - Continuous refactoring of CI
#12) Pervasive Product (Customer) Owners

- Can be a ‘team’ to sort out the demands, but need **a unified decision-maker**
  - Sticky Decisions
- Engaged as a team member
- Outwardly focused toward the market & stakeholder demands
  - Advocate for the team
- Don’t forget:
  - Quality
  - Release planning
  - Tactical vs. Strategic
- Engage the customer!

www.leadingagile.com
#13) Righteous Retrospectives

- For the team only
- Remember Kerth’s “Prime Directive”
  - Everyone tried their best
  - Safe environment
- Lead to risky, high impact changes in approach
- Get the “Elephants” out in the open
  - Sometimes the team needs a hand; external observer
- Be creative – try new things; take some risks
- Challenge each other!
#14) The Power of Complete Transparency

- Opening up your stand-ups & Sprint Planning to everyone
  - Even sales folks and customers
- Rampant Information Radiators
- Celebrate your success AND expose your challenges—with the same emotional
- *Expect engagement* – questions, suggestions, trade-offs towards core goals
- It is what it is…now how do we *ADJUST towards our GOALS*
#15) Ruthless KISS

- Getting LEAN deep into your cultural DNA
- People & Collaboration over Process & Tools
- Fight Gold-plating developing (Just Enough) of EVERYTHING!
- Deliver small increments (Just in Time) and pay attention to feedback
- Comfortable with *little and ambiguity*
- Trust your Product Owner
#16) Testing is Everyone’s Job

- Willingness on the part of the whole-team to pitch in for testing
  - All types, even manual
- Extending it to test automation
- Never letting tests break
- Building in testability
- Listening well to test estimates as part of work estimation
- Understanding functional and non-functional testing
- RCA as a team
#17) The Nuance of a Healthy Backlog

- Considering it a tapestry of work that is considered in turn:
  - Architecture & design
  - Quality & testing
  - Technical debt

- As well, planning
  - Feature workflow & value
  - Dependencies & risk
  - Ultimately deployment

- Product Owner ‘owned’ while empowering the whole team to contribute

- Never ‘done’ grooming; iterative
#18) Strength-Based Teams

- Individuals focus on what they’re good at
  - While still ‘stretching’ themselves
- Notion of Appreciative Inquiry leveraged in retrospectives and continuous improvement focus
- Team-building - interview for strength
- At scale, consider strengths
  - When Release Planning – loading work
  - Load-balancing teams by skill-set
#19) Performing Extraordinary Facilitation

- Grooming meetings – discussions are at the “right level”
- Everyone on the team facilitates
  - Off-line action setting
  - Planning meetings
- Teams get options on the table and pick best solutions
  - Craftsmanship
  - Technical debt
- Business part of decisions; not negotiating, but shared, win-win perspectives
#20) Saying NO as a Leader

- I’ve seen many functional managers “walk away” from their Self-directed Agile Teams
- Knowing when to say…
- Sometimes direction is required
- Putting on the hat of the observer; laying it on the table for the team
- **Courage** to tell it like it is
  - Of Elephants and 800 lb. Gorillas
- Behind the scenes, 1:1 Coaching – towards Agile behavior
Examples...

- Knowing when to step in, step out?
  - When have you stepped in as a leader and realized that it was a mistake?
  - When have you stepped in and realized it was the right decision?

- How do the core Agile Principles help or relate?

- Self-direction…hard or easy? Context-based?
  - What about fostering exploration? Or failure?
#21) Organizational Leadership Alignment

- Senior–Middle–Low level leadership alignment with agile principles
  - Training, operational understanding

- Level A, B, and C leadership present
  - At least one C championing agile

- Projects, PMO, Regulatory requirements, COE aligned

- Asking the right questions of the teams
  - Metrics, status, progress, etc.
Asking the “Right” Questions?

Right

- Quality focused – around done-ness adherence and bug repairs
- Based on information radiators; asking why…?
- In context, for example, at Sprint Reviews regarding features
- Around priority and value of delivered features

Wrong

- Around raw speed; are we there yet?
  - I don’t hear typing / coding going on
- Why do we need to fix that bug?
- Why do we need to refactor that now?
  - Particularly with teams’ who don’t
- Why can’t we cut those points down? What’s so hard about that?
Workshop Wrap-up

- What were the most compelling patterns?
- What essential patterns did I miss?
- Final questions or discussion?

Thank you!
Contact Info

Bob Galen
Principal Consultant,
RGalen Consulting Group, L.L.C.

Director of Agile Solutions,
Zenergy Technologies,

Experience-driven agile focused training, coaching & consulting

Contact: (919) 272-0719
bob@rgalen.com
bob.galen@zenergytechnologies.com
www.rgalen.com

Blogs

Project Times -
http://www.projecttimes.com/robert-galen/

Business Analyst – BA Times -
http://www.batimes.com/robert-galen/

My Podcast on all things ‘agile’ -
http://www.meta-cast.com/