
Scaling Testing in Scrum

Stretching Scrum to Accommodate Legacy & Large-Scale Testing Activity

Bob Galen
Agile Architect, ChannelAdvisor
Principal Consultant, RGCG, LLC
bob.galen@channeladvisor.com
www.rgalen.com bob@rgalen.com

Outline

Introduction

1. Agile Scaling & Scrum
2. The Role of the Product Owner in Scaling
3. Scrum-in-Testing
4. Challenges
5. Q&A

Introduction

Aspects of Agile Scaling

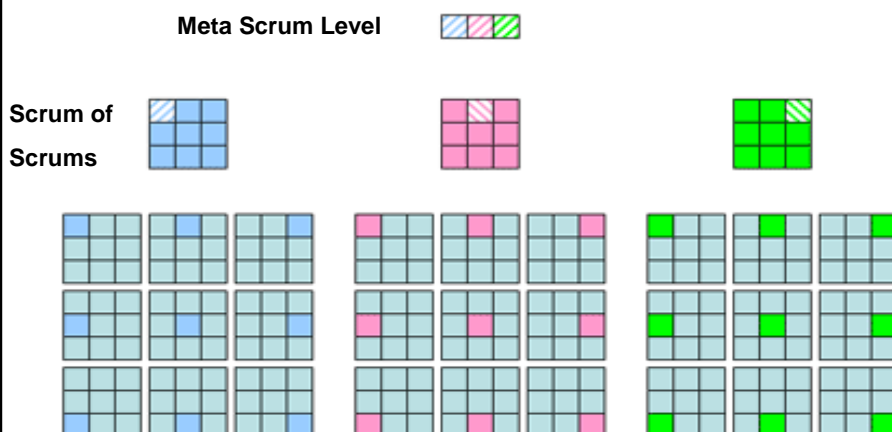
- There are really 4 aspects to Scaling in Agility:
 - **Organizational** – larger teams, distributed teams, outsourced teams
 - **Product** – larger scale projects, interoperability, usability & consistency, and forecasting
 - **Development** – dependencies & integration, varying processes, and cross group collaboration
 - **Testing & Deployment** – regression, regulatory, integration, product maturation visibility, and production deployment readiness
- This presentation focuses on these areas from a testing perspective, but also intersects the other points
- *Of course, in a small, pure agile implementations, much of this is unnecessary and contrary to the basic principles of agility*

Aspects of Agile Scaling

- Industry lessons have lagged in larger scale Agile implementations
 - It's not the "sweet spot" for them and we seem to be mostly on our own
- In the Enterprise, Scrum is leading the way in providing guidance towards scaling, but so far it's sparse in nature –
 - Ken Schwaber will publish an Enterprise Scrum book in June 2007
 - Jeff Sutherland has taken the lead in defining and sharing lessons learned
 - There are still gaps from a Product Owner perspective – although the Certified Scrum PO class should help
 - Large-scale testing implications have been ignored to-date – thus this presentation...

Aspects of Agile Scaling Scrum of Scrums (of Scrums)

Source: Mike Cohn's
www.mountaingoatsoftware.com website.



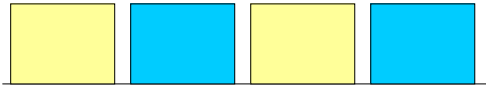
Aspects of Agile Scaling

- And there are roles not defined behind the SoSoS, for example –
 - What are the sorts of conversations & activities that occur at each level? Are there even levels?
 - Who guides the process, tools, & techniques for consistency? Or do you even care?
 - What are the hierarchies behind the SoSoS
 - ScrumMaster(s)
 - Product Owner(s)
 - Lines of business
 - Team collaboration – resource management, allocation, and budgeting
 - Reporting & release coordination
 - Quality, measurement, and governance
- And how do they operate, integrate, and provide consistency w/o command-and-control?

Aspects of Agile Scaling Jeff Sutherland – Parallel Scrum

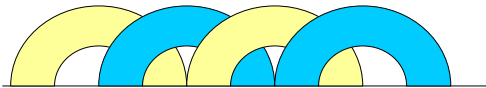
- Sutherland is leading the way toward modifying Scrum towards greater efficiency
 - Scrum levels – A, B, and C
- Sutherland has been using Type C Scrum for 3-4 years at PatientKeeper
 - Sort of the “Nirvana” Scrum state, anyone else at Type C?
- The key point is the organizational change dynamics
 - Transition time compression
 - Forward thinking towards staging & delivery
 - Parallel --- Everything!
 - Organization-wide change implications – including Testing & Quality

Scrum Evolution Parallel Pipelining



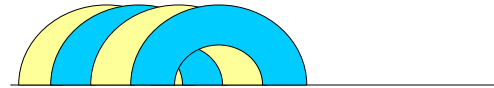
■ Type A Scrum

- Isolated cycles of work



■ Type B Scrum

- Overlapping Iterations
- Backlog continuously refreshed
- Reduced staging times



■ Type C Scrum

- All at once, multiple – simultaneous releases
- Organization of a Meta-Scrum

Evolving Role of the Product Owner

Product Owners Their Evolving Role

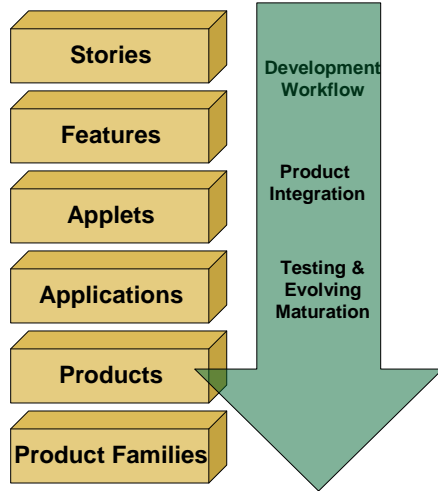
- Within each Scrum, the PO is typically narrowly focused on crafting the Backlog, engaging in progress, and reviewing sprint results
- However, as Scrum scales, the PO team needs to become more focused on:
 - Product Line Evolution – Meta Backlog and coordination across Sprint teams, strategy development & execution, resource load-balancing, and budgeting
 - Cross-Team Planning – SoS coordination, linked goals and backlog work, delivery integration, and staged (forward-thinking) planning
 - Delivery Dynamics – timing, marketing, packaging, interrelationships, customer feedback, and achieving production quality

All of course with the team(s) delivering the product

Product Owners Guiding Testing

- The PO also needs to have a Quality & Testing perspective that within each Sprint focuses on –
 - Developing specific Sprint multifaceted quality release goals
 - Working with the team (Testers) to develop acceptance tests
 - Working with the team to ensure daily convergence towards goals
- Across the SoS:
 - Developing Product & Portfolio quality meta goals that cross all Sprints
 - For example: defect repair targets, test coverage targets, and traceability & reporting requirements
 - Establishing a consistent flow of features
 - Working with the test team on integrating deliverables and qualifying the overall Product via planned Integration & Stabilization Sprints
 - Understanding automation vs. manual testing capacity
 - Focusing everyone towards Product release points

Product Owners Coordination Workflow

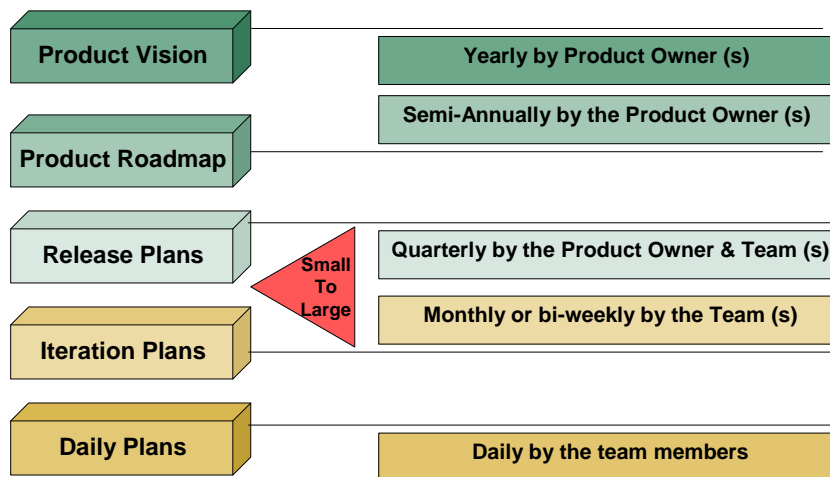


- Scrum of Scrum of Scrums – The PO organization must coordinate
 - Feature timing & workflow
 - Quality & integration workflow
 - Product maturation and release readiness
 - Production deployment

In conjunction with technology and project leadership

While often interfacing to operations and customer facing organizations

Product Owner Planning Levels in Large-Scale Agile Projects



Release Goal Setting A Key for Coordination

- As you scale, each planning level should create criteria (Sprint Goals) that are –
 - Interrelated and cohesive
 - Focused towards the end product release and not simply on each teams deliverables
 - Identify dependencies and overall workflow

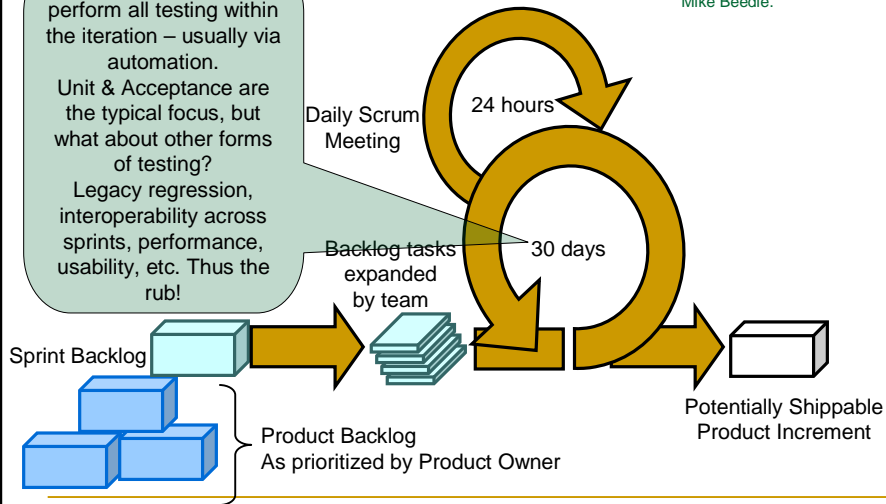
- The traditional notion of Chartering also applies at the higher levels, with Charters defined as:
 - Goals, Objectives & Scope
 - Clearly measurable view to “Done” – Release Criteria
 - Multi-faceted view towards quality (defects, coverage, non-functional requirements)
 - Allowing for team scope trade-offs

Scrum-in-Testing

Process Overview

The Agile intent is to perform all testing within the iteration – usually via automation. Unit & Acceptance are the typical focus, but what about other forms of testing? Legacy regression, interoperability across sprints, performance, usability, etc. Thus the rub!

Source: Adapted from *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle.



v1.1 -- June 2007

RGalen Consulting Group, LLC
Copyright © 2007

17

Inherently Narrow Focus of Agile Testing

Typical Agile Team Testing Focus

- Unit Testing
- Automated Builds – Smoke Testing
- Focused – Customer Acceptance Testing
- Test Driven Development (TDD)
- Very limited – Integration & Regression Testing
 - Focused Towards Automation
 - Limited Exploratory Testing

What's Missing for Larger Scale Organizations?

- Integration Testing
- Functional Testing
- System Testing
- Regression Testing
- Performance Testing
 - Load Testing
- Scenario Based Testing
- User Acceptance Testing (UAT)
- Usability Testing, Other “ility” Testing
 - Exploratory Testing
 - Large-scale Automation

v1.1 -- June 2007

RGalen Consulting Group, LLC
Copyright © 2007

18

Inherently Narrow Focus of Agile Testing

- Early as possible testing – unit level TDD
- Customer engaged in defining acceptance criteria & tests
- Early defect prevention, detection & early repair
- High degree of automation – “everything” runs within the iteration
- Quality as a team responsibility
- Ongoing refactoring as required
- Think of Lean principles as an underlying driver...
 - Just Enough, Just in Time
 - Deliver as Fast as Possible
 - Team integrity & professionalism
 - Holistic, built-in quality

Scrum-in-Testing Transitional Drivers

- High levels of traditional testing
 - Manual regression burden; Legacy systems; Habits across the business stakeholder team; External pressures or expectations
 - PO requires this as part of the Backlog
- Early Scrum implementations
 - Development teams struggle with gaining testing traction – not meeting Agile code quality goals or core practices
 - Testing teams falling into traditional behaviors – artifact based, gatekeeper mentality, and low adaptability or flexibility
 - Insufficient testing resources to staff Sprints and post Development Sprint testing requirements (thinking that Agile teams inherently need *less* testers)

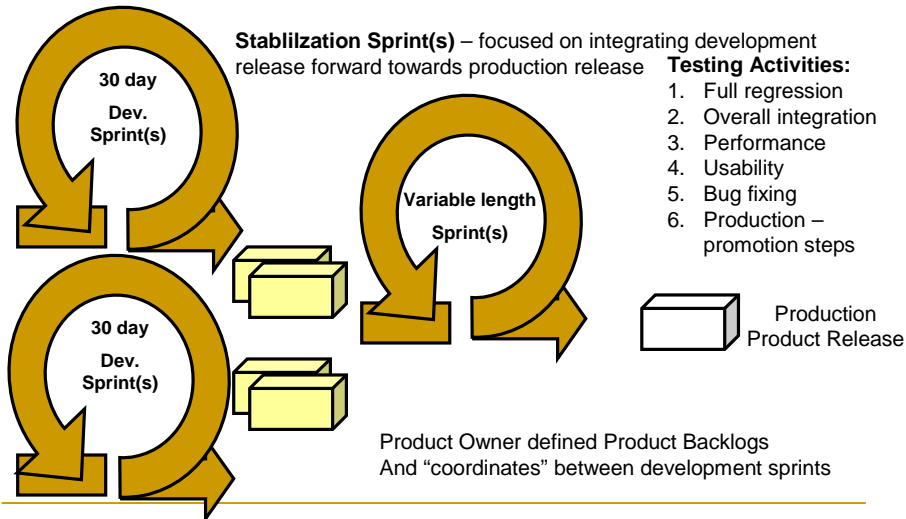
Scrum-in-Testing Coordination Drivers

- Integration
 - Short term integration across (many) Sprinting Development teams
 - Integrating with external data providers
 - Integrating with 3'd parties and outsourcers
 - Equipment limitations & production deployment / promotion models
- The need to give an external team insights into deliverables for including in future work
- PO driven Portfolio & Product road-maps; PMO oriented planning
- Regulatory or compliance requirements (traceability and/or artifact evidence)

Scrum-in-Testing Strategies

- Extending testing activity within the Sprint to include as much coverage as possible:
 - Of course, unit & acceptance for the current iteration features
 - Re-running existing unit & acceptance tests
 - Maintaining existing automation
 - Running partial integration & partial regression testing as possible
 - Cross-team collaborationWhich usually requires a different staffing mix for each Sprint
- Or extending the iterative model to accommodate testing via –
 - Stabilization Sprints
 - Skewed Development & Testing focused Sprints

Scrum-in-Testing Stabilization Sprints



v1.1 -- June 2007

RGalen Consulting Group, LLC
Copyright © 2007

23

Scrum-in-Testing Stabilization Sprints

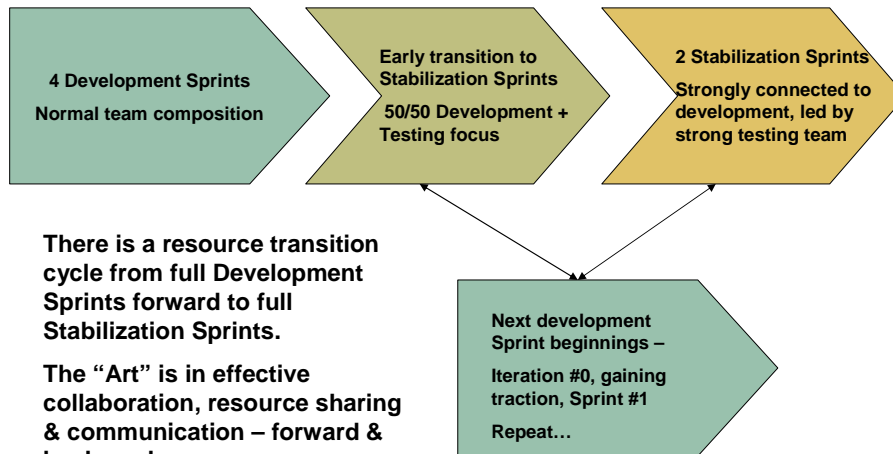
- This is a modified version of the Scrum model where Sprints evolve from ►►►
 - **Note: iteration resource mix changes as you move from development towards stabilization**
1. Pure **Development focused Sprints** – delivering features to PO to...
 2. Early **Integration focused Sprints** – coordinating a building product story and shaking our interoperability dependencies to...
 3. Pure **Testing focused Sprints** – performing more traditional testing activity leading towards production release.
 4. Finally and potentially **Testing Infrastructural focused Sprints** – automation maintenance, next iteration readiness, and customer / usability collaboration

v1.1 -- June 2007

RGalen Consulting Group, LLC
Copyright © 2007

24

Scrum-in-Testing Stabilization Sprint – Sample Workflow



v1.1 -- June 2007

RGalen Consulting Group, LLC
Copyright © 2007

25

Scrum-in-Testing Stabilization Sprint – Content Pressure Release

- Think of Stabilization Sprint(s) as a feature *content pressure release mechanism*. As your content increases, so does its integration risk. You'll want to use them –
 - More often when you have many Development Sprints running in parallel
 - Even using them as an interim integration mechanism – so planning for periodic Stabilization Sprints during your Product Development Sprints
 - In larger organizations, there can be Stabilization Sprint threads running in parallel; creating a need for a SoSoS oriented interactions
- The model typically is used for larger numbers of Development Sprints contributing to a large-scale enterprise product
- Duration and focus can vary from one Stabilization Sprint to the next

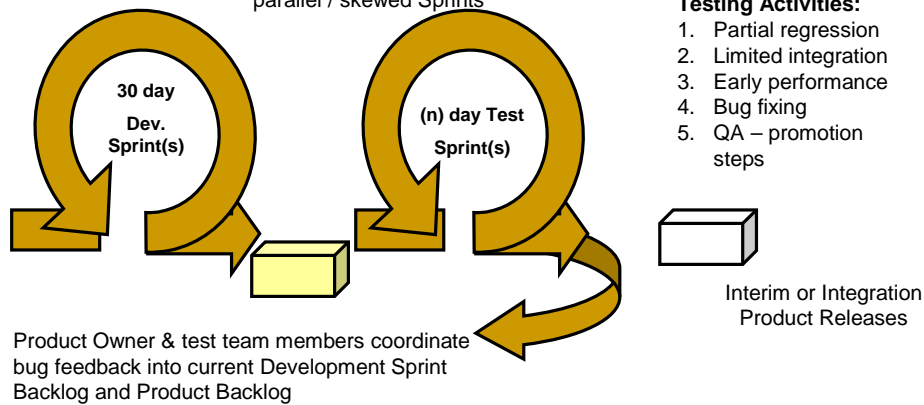
v1.1 -- June 2007

RGalen Consulting Group, LLC
Copyright © 2007

26

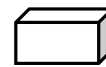
Scrum-in-Testing Skewed Testing Sprints

Skewed Testing Sprint(s) – focused on providing more formal testing feedback by virtually running development & testing in parallel / skewed Sprints



Testing Activities:

1. Partial regression
2. Limited integration
3. Early performance
4. Bug fixing
5. QA – promotion steps



Interim or Integration
Product Releases

v1.1 -- June 2007

RGalen Consulting Group, LLC
Copyright © 2007

27

Scrum-in-Testing Skewed Testing Sprints

- This model balances some traditional testing post Development Sprint against *bogging down* the sprint w/too broad a level of testing
- Usually the testing is focused towards traditional regression and integration testing
 - May also be used for performance, compliance and other non-functional testing activity
- The model typically is used for domains with an existing large scale legacy testing burden (or requiring larger scale testing practices)
 - In this case, the skew allows the development activity to progress while testing is performed
 - Sometimes this is viewed as “Waterfall” testing within Scrum; but becomes necessary if you can’t perform ALL testing within the iteration

v1.1 -- June 2007

RGalen Consulting Group, LLC
Copyright © 2007

28

Scrum-in-Testing Skewed Testing Sprints

- Advantage of handling defects and integration issues close to the originating Sprint
 - Can reserve Sprint Backlog time so that changes can be incorporated immediately in the “next” Development Sprint
- Testing Sprints are usually staffed solely with testers
- At later phases, the testing can turn into more of a Stabilization sprint effort – so a morphing of the two strategies
- Over time as your Agile experience grows, you’ll want to narrow the skew – perhaps with overlapping iterations

Scrum-in-Testing Challenges

Scrum-in-Testing

Typical Challenges – Alignment

- It's important to try and align Scrum iteration tempo across your SoSoS Sprints, putting pressure on your:
 - Cross-sprint Meta Backlog management & planning
 - Sprint Review results & feedback
 - Testing Sprint alignments
 - Dependency management
 - Lab support scheduling & deployment phasing
 - 3rd party integrations

Scrum-in-Testing

Typical Challenges – Resource Balancing

- Resource Balancing
 - Between traditional, development focused Sprints...
 - Including people, equipment, and simply focus
- *Falling Behind*
 - If you skew too heavily towards the traditional, testing stabilization Sprints, you'll lose connection to the next iteration
- *Falling Forward*
 - If you skew too heavily towards the development Sprints, you'll lose the more formal testing & stabilization battle
- More often – teams fall behind when they should be falling forward...
 - Watch out for *Waterfall Testing in an Agile World*

Scrum-in-Testing Typical Challenges – Skill-sets

- Agile skill-sets and collaborative expectations are WAY different!
- Do
 - Define agile testing behavior guideline (role & responsibilities)
 - Encourage pairing and strong collaboration
 - Assess your technical capabilities and begin to aggressively fill in any gaps:
 - Technical domain understanding and direct programming experience
 - Open source automation tools experience
 - Customer collaborative and UAT experience
 - Establish guidelines for balancing between Agility & corporate quality and governance expectations
- Don't underestimate the impact that Agility will have towards your traditional teams' capabilities, approaches and capacity for change

Scrum-in-Testing Typical Challenges – Quality Influence

- Working with the Product Owner (Customer)
 - Becoming a collaborative partner; defining & automating acceptance tests
 - Actively representing the VOC
- It's important to setup clear & broad release criteria for each Sprint and the overall Release
 - Feature goals; usability goals; acceptance confirmation
 - Quality criteria (defects, coverage, types of testing)
 - Process artifact goals (for example: SOX or other compliance, traceability)
- Establishing release readiness (features, quality, interoperability) for PO during Sprint Review (Pass/Fail – goals met?)

Scrum-in-Testing

Typical Challenges – Metrics & Visibility

- As the number of Scrums grow with application size, feedback is gathered more so at the SoS level via –
 - Cross Sprint burndowns and feature tracking
 - Feedback from the testing team in integration issues
 - Product Owner Sprint Review(s) experience
 - Product Roadmap progress – across all of the relevant Sprints
- The Meta ScrumMasters & Meta Product Owners are actively engaged in defining goals and measuring progress against them
- Test teams can / should provide some traditional metrics focused towards –
 - Defects, test coverage & traceability, workflow defect patterns, Sprint release acceptance / goal attainment levels

Scrum-in-Testing

Typical Challenges – Defects

- In pure Agile teams (small teams & projects, quality & testing focused, automation centered) there is little need for traditional defect tracking techniques
 - They test first by developing unit tests and continuously integrating changes;
 - Establish automated acceptance tests and fixing bugs as they're found
- However, in evolving or large-scale Agile teams
 - There is a need for defect coordination across the various product(s) and team(s);
 - Traditional triage, and targeting repairs to specific teams & iterations
 - Product Owner(s) and ScrumMaster(s) are involved in this coordination and delegation
 - Testing teams are at the *center* of these efforts; guiding the teams towards their Sprint & Product Release goals

Wrap-up

- Traditional Scrum (and other Agile methods) struggle to operate in:
 - Non-green field
 - Legacy based or compliance focused
 - Enterprise level or large-scale team environments. This creeps into testing activity as well...

- Testing in Scrum in these contexts should include:
 - Strong partnership with the PO
 - Collaborative development of a strategy that gradually moves from traditional to agile testing
 - Development of a skewed testing model that support PO / Business and Organizational / Quality needs
 - Awareness of the cultural and skill-set changes that need to be made
 - Patience and emergent (Self-directed) change!

Questions?

Thank you!

Agile References

- Augustine, Sanjiv, "Managing Agile Projects", Addison Wesley, (2005)
- Larman, Craig, "Agile & Iterative Development – A Manager's Guide", Addison Wesley, (2004)
- Schwaber, Ken and Beedle, Mike, "Agile Software Development with Scrum", Prentice Hall Publishing, (2002)
- Schwaber, Ken, "Agile Project Management with Scrum", Microsoft Press, (2004)
- Scrum - www.controlchaos.com
- Scrum, Stories - www.mountaingoatsoftware.com, www.scrumalliance.org, www.implementingscrum.com
- Mike Cohn – Scrum of Scrums page/diagram:
<http://www.mountaingoatsoftware.com/scrum/scrumteam.php>
- Jeff Sutherland: <http://jeffsutherland.com>
 - Agile 2005 paper on Type A, B, C Scrums: <http://www.agile2005.org/RP10.pdf>
 - Agile 2005 presentation on Scrum II:
<http://jeffsutherland.com/scrum/ScrumIIAgile2005.pdf#search=%22scrum%20of%20scrums%22>
- Jeff Sutherland Agile 2006 presentation on Distributed Scrum Teams:
www.scrumalliance.org/index.php/content/download/4836/49524/file/SutherlandDistributedScrumAgile2006_v4_28_Feb_2006.pdf

Contact Info

Related ST&P article in June 2007 issue,
www.stpmag.com

Software Endgames: Eliminating Defects, Controlling Change, and the Countdown to On-Time Delivery
published by Dorset House in Spring 2005.
www.rgalen.com for order info, misc. related presentations, and papers.

Robert Galen
RGalen Consulting Group,
L.L.C.
PO Box 865, Cary, NC 27512
919-272-0719
www.rgalen.com
bob@rgalen.com

