Scrum

---------------

It Depends on
Common Sense

http://www.scrumalliance.org
http://www.controlchaos.com
http://www.agilealliance.org
Ken.schwaber@verizon.net
Prerequisites

1. Have read one of the Scrum books
2. Understand the basics of project management
3. Understand requirements and requirements decomposition
4. Have been on or closely involved with a project that builds or enhances a product.
5. Want to know more about how Scrum works, how to use it, and how to implement it in an organization.
6. For anyone with these skills; not just experts.

Purpose

To teach someone how to be a ScrumMaster in their organization.
Agenda:
1. Introduction
2. What is Scrum and Why does it work?
3. The ScrumMaster - Fact and Myth
4. Implementing Scrum - the Basics
5. Teams - Source of Joy and Frustration
6. Scrum - Nuances on the Basics
7. Managing to Optimize Value
8. Implementing Scrum - Advanced Topics and Scaling
9. Closing Topics
10. Graduation

This Agenda, coupled with exercises, will consume the full two days. As appropriate, content may be replaced based on class composition. When of interest to most of the class, student questions will be answered by having the class devise solutions within the framework and principles of Scrum - just as you will have to do at work!
Subject:
Introduction
Why This Course?

Scrum is a simple framework.

This course teaches you to use common sense within this framework. You will be asked to solve your particular situations within this framework.

This course is broad-band communication for learning, as compared to narrow-band, but scalable, techniques such as books.

Through lecture, discussion, exercises and looking at your specific problems, you will learn how to solve your own problems in a Scrum-like manner.

When you are certified, you probably will know more than when you started the course.
Exercise - Scrum Familiarity

Have participants get on their feet (if necessary, clear an area of the room); Define an imaginary line running across the room on which participants will place themselves)

Exercise 1: How effective are the existing processes and development practices within your organization?
10 == couldn't be better; living the dream!
1 == failing on multiple levels; train wreck waiting to happen

Exercise 2: How familiar are you with Scrum?
10 == read both books; I know what it's about; I've practiced it!
1 == seemed interesting so I decided to take a training course

Exercise 3: What development model best describes your project working environment?
10 == consulting: firm is working with external clients; client-driven development
1 == corporate: managing development of in-house product or service

Have participants answer this question by arranging themselves on the continuum. "Arrange yourselves on the continuum in keeping with your answer to this question"

Once arranged, have participants look around and notice where they are relative to all of their coworkers. Once continuum has "formed", provide enough time for people to look around.
At start was 80/20 external to internal developers

Currently ratio has reversed to 20/80

Scaling # of instructors
Subject:
What is Scrum and why does it work?
Origins of Scrum

The New, New Product Development Game*

Lean

Iterative, Incremental Development, Timeboxes

Smalltalk Engineering Tools

Scrum

Exercise – Defined Process

1. Form pairs.

2. Assign one person the boss, the other is the worker.

3. The boss can give the following commands – Go, Stop, Right, Left, Faster, Slower

4. The worker must follow the boss’s commands.

5. The boss is responsible for having the worker proceed 60 normal paces within two minutes, from the time “go” is said, until “stop” is said, by the moderator.

6. The boss can command the worker but not touch the worker.
Exercise - Empirical Process

1. With the same teams as before, except everyone is a worker and responsible for figuring out how to proceed during the exercise by him or herself.

2. Each team proceeds 60 normal paces within two minutes, from the time “go” is said, until “stop” is said, by the moderator.
Categorization of complexity in development projects

- People dimension adds another level of complexity
- Last simple project was in 1969
“It is typical to adopt the defined (theoretical) modeling approach when the underlying mechanisms by which a process operates are reasonably well understood. When the process is too complicated for the defined approach, the empirical approach is the appropriate choice.”

*Process Dynamics, Modeling, and Control, Ogunnaike and Ray, Oxford University Press, 1992*
Scrum Overview

• Empirical management & control process
  - inspect and adapt feedback loops;
• Used to manage complex projects since 1990;
• Delivers business functionality in 30 days;
• Scalable to distributed, large, and long projects;
• CMM Level/3 and ISO 9001 compliant;
  and,
• Extremely simple but very hard.
Companies Using Scrum

Bottom Up

Microsoft, Sun, Sammy Studios, Siemens, CNA, State Farm, State Street Bank, Philips, BBC, IBM, SAIC, LMCO, APL, Ariba, Federal Reserve Bank, HP, Medtronics, Motorola, TransUnion

Top Down

IDX, Siemens Medical, Gestalt, Wildcard Systems, Primavera, Yahoo, Conchango, BMC, Lexis-Nexis, Bentley Systems, Bose, CapitalOne, Federal Reserve Bank, ClearChannel, Xerox

??

French Post Office, SAP, etc.
Exercise: The Art of the Possible

Explore the difference between planning a party if every sentence begins with “yes, but” and “yes, and.”
Timeboxes, Roles, Rules
Sprint

1 day Sprint planning meeting (4hr/4hr)

1 day Sprint review and retrospective (4hr/4hr)

Development Work

30 day Sprint at sustainable pace
Scrum Phases

Scaling
Planning        Developing        Implement
What is Scrum

Scrum is a way of harnessing creativity, the joy of work, the pleasure of teamwork into extraordinary productivity in building complex products.

Or, is it a way to build complex products that enables creativity, collaboration, and being alive?
Subject:
Implementing Scrum - The Basics
Project QuickStart

www.controlchaos.com/quickstart.pdf

Participants: Team, Product Owner, ScrumMaster, chickens, teacher

Agenda

1. Teach Scrum concepts, theory, practices
2. Present project vision, goals, timelines
3. Teach Sprint planning
4. Define Product Backlog for at least three months
5. Brainstorm about overcoming impediments
6. Brainstorm about Product Backlog for next Sprint - team commits
7. Team defines Sprint Backlog
8. Teach daily Scrum, Sprint review, Sprint signature, and management
9. Discuss engineering tools and practices
Agile Skeleton

Product Backlog:

Figure 2

Doing the Right Thing

• Easy to implement within 1 day
• Improves ROI
• Solves customer involvement
• Removes floundering and politics
Pair Dialogue: Skeleton

Pair with someone you don’t know. Turn to each other and share short answers to the following for 5 minutes:

1. Could you implement Scrum in two days as described?

2. Would you have to tell anyone or gain anyone’s permission? If yes, what would you tell them?

3. What problems would you encounter?
Scrum Scenario 1: Helping Adidas Get Agile

Adidas Shoes, Inc. is evaluating software consulting firms to take over development of Shoe-In, a browser-based web application that provides a searchable product catalog, order placement and tracking services to corporate customers like Foot Action, Oshmans, and Academy Sporting Goods. Originally designed to reduce operating costs and prevent buildup of unused inventory, the Shoe-In application has exceeded management’s wildest expectations. Since deploying v1 of the product 9 months ago to a small set of corporate customers, the application has saved the Running Shoe division more than $165 million and increased sales by more than 23%. Adidas Decision to Outsource Noting the heroic effort required to develop and deploy the v1 product, and the technical support challenges facing the makeshift development team, Adidas executives have agreed to outsource development of the v2 product to a dependable development firm. Drawn by claims of improved productivity, higher quality, better ROI and greater customer satisfaction, Adidas is evaluating consulting firms that utilize Agile development methods to manage product development.

Your Mission: Dynamic Development, Inc. is a small (5-person) but highly regarded software consulting firm specializing in Scrum. As co-founder and CTO of Dynamic Development, you will be meeting with Marvin Moneybags, executive sponsor of the Adidas Shoe-In product to:

1. Explain how your firm will approach the project using Scrum
2. Hear and address Marvin’s questions and concerns
3. Help Marvin understand why he should hire your firm.

Your mission is to land the Shoe-In project. Marvin knows everything about his business but very little about software development. Marvin’s previous experience with an outsourced development project ended in disaster: the planned 8-month, $2 million project was delivered in 18 months at a cost of over $5 million.

Company Background Adidas is a hard-driving business-focused company in the highly competitive athletic shoe market. The executive team is driven by results and sees the enormous potential of Shoe-In to continue to reduce operational costs and to drive corporate business. Corporate customers, executives, marketing, and customer service are all demanding business-critical features for the next release, which for management can’t be soon enough.
Exercise: Risk

You are working at FatBurger and are the only person on duty. A customer approaches and orders a Double Fatburger Deluxe, with onions, cheese, and bacon.

You ring up the order. The price is $5.65.

The customer informs you that he only has $1.20

What do you do and what do you tell the customer?
## Key Roles and Responsibilities

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product Owner</strong></td>
<td>- Defines the features of the product, decides on release date and content &lt;br&gt;- Is responsible for the profitability of the product (ROI) &lt;br&gt;- Prioritizes features according to market value &lt;br&gt;- Can change features and priority every 30 days &lt;br&gt;- Accepts or rejects work results</td>
</tr>
<tr>
<td><strong>ScrumMaster</strong></td>
<td>- Ensures that the team is fully functional and productive &lt;br&gt;- Enables close cooperation across all roles and functions and removes barriers &lt;br&gt;- Shields the team from external interferences &lt;br&gt;- Ensures that the process is followed. Invites to daily scrum, iteration review and planning meetings</td>
</tr>
<tr>
<td><strong>Team</strong></td>
<td>- Cross-functional, seven plus/minus two members &lt;br&gt;- Selects the iteration goal and specifies work results &lt;br&gt;- Has the right to do everything within the boundaries of the project guidelines to reach the iteration goal &lt;br&gt;- Organizes itself and its work &lt;br&gt;- Demos work results to the Product Owner</td>
</tr>
</tbody>
</table>
Exercise

Portal Product – send in to straighten out engineering team because they haven’t delivered in 7 months.

Five product owners, each with their own P&L.

What would you do to improve the situation?

Exercise

Sprint Planning meeting where David is Product Owner. What do you do?
Exercise - Plan to a Vision

MLB has invested $554m in getting a law passed requiring the use of MLBTix on March 30 and having the requirements developed.

It is now December 7 (4 Sprints away) and Bud Selig needs your team to build MLBTix so it will at least allow people to buy and sell tickets. Otherwise the law and investment is null and void. Funding is virtually unlimited, but entangling partnerships are unlikely to be approved.

Bud wants to know: can your team build this skeletal system for MLB? The Commissioner’s marketing organization wants the Commissioner to announce MLBTix at a press conference on January 15. Do you concur, and what should the Commissioner announce?
Read this Project Background

Today’s date is December 7, 2003. You have been selected to be part of a team for a complex project with a compressed schedule. Although the general nature of what is wanted is known, the specific functionality that will implement it will have to be determined as the project progresses.

Background:

Overall attendance at baseball games has increased over the last ten years. In some cities, such as Boston, almost all games are sold out and obtaining tickets through normal channels is nearly impossible. Major League Baseball (MLB) rules prohibit the resale of tickets at a profit. Scalping is illegal and has been recently limited. The primary distribution channel for buying tickets is eBay. Although all auctions for tickets on eBay are supposed to be capped at the retail price plus expenses, MLB has learned that, through a variety of workarounds, these tickets are being scalped for prices of up to 1000% of the retail price.

Project:

The MLB Commissioner’s office has commissioned a project to control the resale of baseball tickets. Through recent legislation, as of the 2004 baseball season, all ticket resale can only take place through facilities authorized by MLB. MLB has decided to develop such a facility solely for its own purposes, through the presence of a dedicated website. The site will be known as MLBTix.

Through functionality similar to eBay, but specific to MLB, buyers and sellers will be able to sell and buy tickets online. Sellers will auction the tickets to the highest bidder through an auction capability. The seller sets an initial bidding price of their own choice without floor or ceiling conditions established by MLBTix. The seller can also limit the duration of the auction, setting a start and end date and time. If the ticket(s) are successfully sold, the buyer pays the seller through MLBTix credit card facilities. Then the seller will mail or express the tickets to the seller. MLBTix will have a facility for the buyer to notify it when the tickets have been received, at which time MLBTix will mail a check for the proceeds (less 25% MLB fee that is deducted) to the seller.

The Commissioner will be announcing MLBTix at a news conference on January 15. He hopes for some functionality to be available by opening day on March 30, 2004, and for the site to be fully functional by the All Star break on July 18, 2004. The anticipated release schedule is:
1. March 30, 2004 – MLBtix site is up. Buyers and sellers can register. Sellers can make tickets available at a fixed price, which buyers can pay in full via credit card. MLBTix is a middleman, all transfer of tickets is between buyer and seller. MLBTix receives 25% commission for all transactions.

2. June 30, 2004 – same as March 30 release except full-functioning auction capability is present.

3. August 30, 2004 - same as June 30, except buyers are able to get groups of collocated tickets, view the locations in parts, check inventory.

Funds for the project are ample and should not be considered an unreasonable constraint. The date and functionality are the deliverables. Facilities or packaged software to support MLBTix can be either bought or developed, whichever supports meeting the date. The Commissioner needs a heads up on the likelihood that the MLBTix will be available by the above dates prior to his press conference.
Functional Requirements:

- Register as a potential seller of tickets and be assigned a userid and password.
- Register as a potential buyer of tickets and be assigned a userid and password.
- Maintain a profile under the userid, including email, addresses, preferences, and credit card information.
- Place tickets up for auction, declaring a floor price, start of auction time/date, and end of auction time/date. Indicate sufficient information so that buyers can ascertain that the tickets meet their requirements (for the right days, right teams, right number of seats located next to each other, and the seat locations in the ball park).
- Conduct an auction for the tickets to registered buyers.
- Successfully conclude the auction by awarding the tickets to the highest bidder by the end date and, at the same time, debiting the buyers credit card and placing the funds in a MLBTix account.
- Notifying the seller of the successful sale of the tickets and the delivery information for the buyer.
- Providing the buyer with a mechanism for indicating that the tickets were not successfully received by the selling date plus a specified period of time (a week?).
- Transferring the funds for the ticket sale less 25% to the seller at the end of the specified delivery time, unless the buyer has indicated otherwise.
- Transferring the 25% plus any interest to a corporate MLB account from the MLBTix account automatically.
- Providing inventory and inventory search capabilities for teams, tickets, dates, and seats within park.
- Providing for promotions on MLBTix.
- Ability to identify and ban abusers of MLBTix.
Nonfunctional Requirements:

- 250,000 simultaneous users with subsecond response time.
- Secure for the level of financial activity envisioned (2,000 tickets per day at an average selling price of $50).
- Scalable to 1,000,000 simultaneous users as needed.
- 99% availability 24x7.

Development Context:

1. A development environment for building Microsoft .Net products is ready for you. The system will be built using Intel technology and .Net software running on SQL Server.
2. The development team members all live within easy commuting distance of the development site.
3. There are currently cubicles in the development site.
4. The development environment is wireless and has all power and networking capabilities already operating.
5. The development environment uses Microsoft development tools such as Visual Studio.
6. You are required to use a source code library, check in code every time it's changed, built the software at least daily, and unit and acceptance test the software every time that it is built.
7. Scrum will be used as the development practice. Aspects of Extreme Programming or any other engineering practices, such as coding standards, are up to the team.
8. All of the developers have excellent engineering skills, but they have only heard of Scrum and Extreme Programming, or used them sparsely so far.
9. The team consists of all development engineers with excellent design and coding skills. However, they are still responsible for all testing and user documentation. They may acquire contractors to assist with this. The engineers on the team average 10 years of progressive experience on software projects using complex technology and Microsoft products.
10. All team members are baseball aficionados.
11. A QA environment already exists.
12. There are no adequate testing tools, continuous build tools, refactoring tools, and VSS is perhaps not adequate for the job.
Subject:
Implementing Scrum - The Basics (continued)
Building the Thing Right

• More time to implement
• Solid engineering practices
• Solid engineering infrastructure
• XP
Exercise: Sprints

1. Is there such a Sprint as an “analysis” Sprint where requirements are pulled together?

2. Is there such a thing as a “testing” Sprint?

3. What is a “stabilization” Sprint and what should be done with it?

4. If a project requires a lot of infrastructure and architecture work that will take eight weeks to complete, should the first Sprint be eight weeks long? Is the architecture an adequate deliverable?
Optimized Engineering Practices
Transitional Engineering Practices
Scope of “Done” Changes

What is “done?”

Planning
Analysis
Design
Coding
Testing
Performance
User Acceptance
Pilot
Live

Architecture, Infrastructure
Extend the definition to include all development.
Pair Dialogue: Defining “Done”

Pair with someone you don’t know. Turn to each other and share short answers to the following for 5 minutes:

• What does “done” mean in your current project?’

• What issues do you see with this definition of done?

• How would you address them?

• What engineering problems do you see with this approach?

• How would you rectify them?
Done

At a team or team of teams level, this means that common engineering infrastructure and tools are probably being employed, such as:

• Version control
• Build
• Testing
• QA Environments

When this is not true, the manual overhead to ensure “done” increases substantially and often cannot be ascertained.
Multiple Team Integration

Teams separated along functional lines

Team 1  Team n

User Interface Layer

Business Logic Layer

Persistence Layer

One code base, continuous integration, work divided by functionality
Multiple Team Integration

Teams separated along layers

One code base, continuous integration, work tested at integration layer and integration bugs passed back for immediate fixing.
“Done” with Unsynchronized Technology

Sometimes testable integration cannot be obtained across all levels. “Done” may be different at some levels, such as Infrastructure, Hardware, Firmware development, etc.

In these cases, build stubs or simulation layers that conform to frequently confirmed specifications. As frequently as possible, create probable, testable “done” product.

In the meantime, update the engineering practices in those areas so that they can create “done” product more frequently.
Done with Core Product, Infrastructure and Common Facilities

• Few people with expertise in the area
• Can include common engineering tools such as application frameworks
• Fragile, inadequate test harnesses, entangled code
• Low velocity when being worked on
• Has its own product backlog
• Product backlog contains technology improvements, refactoring, rebuilding.
• Product Owner is often a senior engineer
Burndown Chart … the velocity of turning requirements into potentially shippable increments of functionality.

The core functionality line shows the impact of continued poor quality practices from the era of opacity. Although poor quality may be necessary at the start of a company, it is not a sustainable practice.
As functional teams need work done on infrastructure, they inject those items in the infrastructure product backlog. As dependencies arise, the infrastructure team is more fully staffed with developers from the functional teams until the upgrades are done.
Scope of “Done” Changes – Cross-Functional Process for a Team

Cross-functional iteration
development process

- Analysis to design acceptance tests
- Build functionality
- Build test materials
- Build documentation
- Confirm that functionality works as designed
Subject:
The ScrumMaster – Fact and Myth
Scrum

Most projects deliver software every 6 to 18 months. Scrum reduces this to many 1 month deliveries to increase control via inspect/adapt.

This puts stress on the team and organization, exposing underlying problems and limitations.

The ScrumMaster’s job is to prioritize these problems and help the organization overcome them to get better at software development, managing software investments, and becoming a community to work in.
As a ScrumMaster, you are responsible for:

Removing the barriers between development and the customer so the customer directly drives development;

Teaching the customer how to maximize ROI and meet their objectives through Scrum;

Improving the lives of the development team by facilitating creativity and empowerment;

Improving the productivity of the development team in any way possible; and,

Improving the engineering practices and tools so each increment of functionality is potentially shippable.

Leader and Facilitator
As a ScrumMaster, you will have to contend with:

- The tyranny of the waterfall;
- The illusion of command and control;
- The belief in magic; and,
- The era of opacity.
Pair Dialogue: Multiple Roles

Pair with someone you don’t know. Turn to each other and share short answers to the following for 5 minutes:

1. What are some of the challenges a ScrumMaster would have if he/she was also a member of the team?

2. What would help you overcome these challenges?
Scrum and the ScrumMaster

Scrum is a simple iterative, incremental skeleton with some rules.

Equipped with a resolute, patient ScrumMaster, Scrum can be used to transform software development into a profession, projects into valuable endeavors, and development organizations into communities that people look forward to working in.

It takes time. Remember that a dead ScrumMaster is a useless ScrumMaster.

Remember that Scrum is just a framework. It doesn’t fail. Sometimes people can’t stand what it exposes.

ScrumMasters are the key to its degree of success in transforming organizations.
Within the skeleton and heart of Scrum, using Sushi delivery, any management practice to scale a project can be applied. Bad practices will be visible as impediments. Good practices will be invisible.

It Depends on Common Sense
Day in Life of ScrumMaster

• Ensure everyone is doing what they have agreed to do;

• Determine where Scrum is compared to where it could be and update your own Scrum product backlog;

• Work the product backlog; and,

• Use all of your senses, including common sense, and remember that you have no authority.
To listen fully means to pay close attention to what is being said beneath the words. You listen not only to the 'music,' but to the essence of the person speaking. You listen not only for what someone knows, but for what he or she is. Ears operate at the speed of sound, which is far slower than the speed of light the eyes take in. Generative listening is the art of developing deeper silences in yourself, so you can slow our mind’s hearing to your ears’ natural speed, and hear beneath the words to their meaning.

— Peter Senge
Subject:
Teams - the source of productivity and creativity and frustration
Agile Heart

• Let people figure out the right thing to do, and then do it.

• Let people be creative.

OOPSLA'02

Doing the Right Thing the Right Way

• Hardest to implement
• Improves productivity
• Work becomes a pleasure
Scrum Teams
  Forming
  Storming
  Norming
  Performing
Scrum Teams

• Self-organizing;
• Cross-functional with no roles;
• Seven plus or minus two;
• Responsible for committing to work;
• Authority to do whatever is needed to meet commitment;
• Open, collocated space;
• Resolution of conflicts;
• Rules of Etiquette:
  • Never use the word "you" because the other person may feel on the spot and defensive.
  • Never refer to history (e.g., "three months ago, you said...!").
  • Be on time for meetings; if you are late, apologize and pay a late "penalty."
  • If everyone is talking at once, use a pen to determine who talks. Whoever is holding the pen talks, everyone else listens.
  • Everyone's opinion is important and needs to be understood and taken into account.
  • No name calling.
Exercise: Teams

You are the ScrumMaster and are heading for the team room. The functional analyst runs past you crying and the lead engineer runs past you enraged, both on the way to their functional managers' offices.

You go into the team room. You can cut the tension with a knife it is so thick.

Apparently, the analyst has been writing specs and giving them to the engineers, who then change them as they see fit. Anger over this has been building for three weeks.

What do you do?
Exercise: Teams

A team is in its Sprint planning meeting. Half of the team is from a technical company that is running the project. The other half consists of contractors.

A usability engineer from the technical company demands to know how the contractors will ensure adequate user interface. A system architect from the technical company demands to know how the contractors will follow and not ruin the systems architecture they define.

What is wrong here? What do you do?
Exercise: Teams

You are the ScrumMaster. Everyone on the team except John meets with you. They tell you that John is not doing his work, is offensive, is difficult to work with, and they want you to fix the problem.

What do you do?
Exercise: Teams

You are the ScrumMaster at the first Daily Scrum. There are two programmers, a tech writer, and two quality assurance people.

The programmers report that they were in a design meeting and will continue today. The tech writer says that they are working on the table of contents. The quality assurance people report that they are setting up the test bed?

You ask the tech writer and QA people why they aren’t in the design meeting. They say they weren’t invited. You ask the programmers why they weren’t invited. They ask you what possible benefit these people would add to design?

What do you do?
Exercise: Teams

Collocation means the placement of the entire team together so they can see, hear, and work with each other without moving.

1. Do you have such space right now?
2. If not, what do you tell the teams?
Pair Dialogue: Scrum Team

Pair with someone you don’t know. Turn to each other and share short answers to the following for 5 minutes:

1. What are some of the challenges if everyone on the team doesn’t feel that they are on the team?

2. How do you make everyone feel as though they’re on the team?
Engage Human Resources

Change from specialists to developers

• Career paths
• Self-managing teams
• Instruction in conflict resolution and team building
• Awareness of abusive situations
• Change in compensation to team-based
• General awareness of change
The 1st Scrum Project Is An Odd Duck
A Scrum team only consists of the developers, product owner, and ScrumMaster. Nobody else is on the team.
Initiating Scrum Projects - Funding

Plans

Product Owners

Capitalization Committee

Funding
New
Projects,
Releases

Product Owners with funds, plans and product backlog

MetaScrum

Copyright 2003-2005, ADM, All Rights Reserved v6.0
Capitalization committee maintains prioritized list of projects, releases, initiatives based on return on investment or some other value.

<table>
<thead>
<tr>
<th>Name</th>
<th>Plan</th>
<th>ROI</th>
<th>Desired Start</th>
<th>Desired End</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proj1</td>
<td>Plan1</td>
<td>28%</td>
<td>1/2/06</td>
<td>4/1/06</td>
</tr>
<tr>
<td>Proj2</td>
<td>Plan2</td>
<td>26%</td>
<td>10/3/05</td>
<td>1/2/06</td>
</tr>
</tbody>
</table>
MetaScrum is responsible for staffing projects to maximize productivity and cross-functional capabilities, minimizing centers of excellence, and maximizing team focus.
Productivity drops as team members project switch
Done

When team members report to each other, and when teams report to the product owners, the ScrumMaster is responsible to ensure that the definition of “done” is known and adhered to.
Scrum of Scrums ... at as many layers as actionable information is needed. Warning ... too much information and too frequent meetings will result in overaction and floundering.
Exercise

What is the purpose of a “weekly” daily Scrum?

What is the purpose of a “monthly” daily Scrum?

When might they be required and how would they be implemented?
Done

When people from multiple teams report at a Scrum of Scrums, or present at the Sprint Review, the ScrumMaster is responsible to ensure that the definition of “done” is known and adhered to.
ScrumMasters for Scrums of Scrums of Scrums ...are responsible for having:

• An identifiable Product Backlog (superset of lower Product Backlogs)

• A Product Owner and team (team consists of people from lower level teams)

• A Sprint Planning and Sprint Review that decomposes to or comprises lower levels of Sprints.

• Daily Scrums

Overall, this ScrumMaster is responsible for maintaining the overall Scrum process for that level of rollup or control.
Project Managers are a source of team ScrumMasters

Functional Managers are a source of Scrum of Scrum ScrumMasters

Exercises

What are the problems that these people might encounter in fulfilling their ScrumMaster duties?

What are other sources?

What is the source of functional expertise, standards, and guidelines?
Scaling Recommendations

1. Correlate team organization to subsystems or modules with minimal cross-over.

2. Implement development infrastructure to support number of developers and location of developers so it acts as a single development environment.

3. Implement meeting and communication infrastructure optimized for number and location of teams.

4. Develop standards, guidelines, training courses, templates, and frameworks to minimize the coordination required for intended scaling.

5. Develop coordination mechanisms for multiple teams.

6. Ensure each team has sufficient resources. Carefully consider shared resources.

7. Implement ways to develop a common culture across teams.
Multi-team or offshore Development

• Martin Fowler observed at a recent workshop (Canadian Workshop on Scaling XP/Agile Processes, Banff, Alberta, Canada, February, 2003), “scaling agile projects is the last thing you should do.”

• Agile is optimized for a single team of developers working in an optimized environment

1,000,000 lines of C++ code..........BWP..........Industry standard
Time in months.............................31................>50
Staff.........................................8.............>100
Function points per staff month......77.............2

Dispersed Development

1. Xerox, Pearson Educational
   (Sub-Contractor model)
   1. Development of requirements, specifications, architectures, and design at business
   2. Development of code offshore.
   3. Use iterative, incremental development offshore, monitored by the two inspect/adapt cycles (Sprint Review, Daily Scrum)

2. ThoughtWorks
   (Single, Virtual Team Model)
   1. Build an infrastructure such that everyone is an equal employee and location is irrelevant.
   2. Spend some savings to ensure people know and work with each other.
Dispersed Team Recommendations

• Co-locate team as often as possible, especially at inception and key milestones. Rotate members around.
• Invest in (and plan for) tools that provide a shared environment. Plan to experiment.
• Establish a single global instance of project assets, easily accessible by all.
• Try virtual team building (team wiki w/bios & photos).
• Establish known hours, with as much overlap as possible.
• Apply high cohesion and low coupling to allocation of work to sites.
• Develop a shared team vocabulary.
• Don’t let anyone go dark.
• Apply Scrum-of-Scrums concept when mass remote meetings unproductive.
Staging Scalability Requirements

Figure 3
Phasing Scaling Work

Initial Product Backlog
- Functional requirement
- Non-functional requirements
- Staged, scalability requirements
- the rest of the functional and non-functional requirements

Product Backlog
- Functional requirements
- Non-functional requirements

Single Team

Many Teams
Developing Architecture

• Architecture and infrastructure are high priority non-functional requirements

• Must be completed to prove that functional requirements can be implemented satisfactorily

• Every Sprint still must deliver at least some piece of business functionality

  • To prove that architecture or infrastructure works

  • To prove to customer that work they care about is taking place
Cost allocation changes within the project as architecture and infrastructure are put in place.

![Relative Costs Diagram](attachment:image.png)

The diagram shows the relative costs over sprints, with costs allocated to infrastructure and architecture and functionality.
Inter and Intra Project Dependencies

• **Within the Sprint cycle**
  
  • Have the dependency commit to your team’s Sprint goal.
  
  • Most of these dependencies are from “centers of excellence” and should disappear across time.

• **Outside the Sprint cycle**
  
  • Use prioritization of Product Backlog to synchronize dependencies; readjust as necessary.
Exercise

A large commercial software company has 500 developers and engineers in the US and Europe. They also have 250 employees in Bangalore, India who do nothing but QA. The two groups have trouble working together.

As a part of implementing Scrum, how might you straighten this situation out?
Scrum 59 min - Overview

• How does the exercise works?
• Sprint Goal: Develop a Brochure in a 3 day Sprint
• Complete Sprint Planning Meeting - 10min
  - At least 5 Product Backlog Items (select any regardless of priority)
  - 2-3 Sprint Tasks per Item
• Conduct a Sprint Day 1 - 8 min + 4 min Daily Scrum
• Conduct a Sprint Day 2 - 8 min + 4 min Daily Scrum
• Conduct a Sprint Day 3 - 8 min
• Conduct a Sprint Review and Demo - 13 min
Doggy Daycare Brochure Backlog

- Create cover art, brand, and/or logo
- Define major care sections
- Define “Ultra Doggy Spa” service
- Outline boarding options
- Write testimonials
- Define all service offerings
- Set pricing structure for services
- Suggest daypack contents to accompany clients
- Outline full week lunch menu
- Complete a guarantee policy
- Provide satisfied customer testimonials
- Complete a certification structure
- Outline minimum requirements (shots, temper, breeding, etc.)
- Complete bios on staff members (backgrounds, training, interests)
- Define discounted partner pet services
- Contact Information

10 min - Sprint Planning
8 min - Day 1 of Sprint, 4 min - Daily Scrum
8 min - Day 2 of Sprint, 4 min - Daily Scrum
8 min - Day 3 of Sprint,
13 min - Sprint Review
3 ... 2 ... 1 ...

GO!
Close of day

• Skills Inventory

• “Make sure we cover...” list
Subject:
Scrum – Nuances on the basics
Scrum Flow

Timeboxes, Roles, Rules

Copyright 2003-2005, ADM, All Rights Reserved v6.0
Meetings

- Sprint Planning
- Daily Scrum
- Sprint Review
Sprint Planning Meeting

Product Backlog

Team Capabilities

Business Conditions

Technology Stability

Executable Product Increment

Review, Consider, Organize

Next Sprint Goal

Product Backlog

Sprint Backlog

2 - 4 hours meetings ... time boxed
Sprint Planning Meeting

Team Capacity

Product Backlog

Analyze, evaluate and select Product Backlog for Sprint

Estimated Work

Decompose to specifications and tasks, estimate tasks

Budgeted Work in Tasks
Exercise: Sprint Planning Meeting

The first part of the Sprint Planning meeting is time-boxed to four hours. The Product Owner presents the probable Sprint Product Backlog to the team.

1. What if the Product Owner doesn’t show up?
Exercise: Sprint

The team is focused on fulfilling the commitment that it has made to the product owner. However, a critical new business requirement has come up on the 17th day of the Sprint. As the Product Owner, what do you do?

Describe PlanetOut Partners

1. How do you get one Product Backlog?
2. What do you do when 3hr 55m is used up?
Sprint Backlog

- Tasks to turn product backlog into working product functionality
- Tasks are estimated in hours, usually 1-16
- Tasks with more than 16 hours are broken down later
- Team members sign up for tasks, they aren’t assigned (be patient, just wait!)
- Estimated work remaining is updated daily
- Any team member can add, delete or change the Sprint Backlog (theirs or new)
- Work for the Sprint emerges
- If work is unclear, define a Sprint Backlog with a larger amount of time ... break it down later.
- Update work remaining as more is known, as items are worked
### Sprint Backlog

If the team believes that this is too much, they can meet again with the Product Owner

Copyright 2003-2005, ADM, All Rights Reserved v6.0
Monitor the task board

<table>
<thead>
<tr>
<th>Story</th>
<th>To Do</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;A user can...&quot;</td>
<td>Code the...</td>
</tr>
</tbody>
</table>

Tests
Ready

5
8
A sample task board
Another sample task board
Sprint Backlog Estimates

- Work remaining reporting during a Sprint updates the estimated number of hours required to complete a task.

- This should not be confused with time reporting, which is not part of Scrum.

- There are no mechanisms in Scrum for tracking the amount of time that a team works.

- Teams are measured by meeting goals, not by how many hours they take to meet the goal. Scrum is results oriented, not effort driven.
Scrum Practices -Sprint

- Thirty calendar day iteration
- Team builds functionality that includes product backlog and meets Sprint goal
- Team self-organizes to do work
- Team conforms to existing standards and conventions
- Abnormal termination of Sprint
Sprint

• If the Sprint requires >20% more work during the Sprint than was planned by the second day after the Sprint Planning meeting, it needs to plan better.

• If team members report the same item more than one day, they need to plan better and decompose the tasks to a greater level of granularity.

• The Product Backlog estimate is a mutually agreed to budget, even if it is a SWAG. If the team is going to exceed the budget, it needs to escalate the decision. Otherwise, the team introduces scope creep.

• Every member of the team is responsible for managing the team.
Sprint Abnormal Termination

• Sprints can be cancelled before the allotted thirty days are over;

• Team can cancel Sprint if they feel they are unable to meet Sprint goal;

• Management can cancel Sprint if external circumstances negate the value of the Sprint goal; and

• If a Sprint is abnormally terminated, the next step is to conduct a new Sprint planning meeting, where the reason for the termination is reviewed.
Daily Scrums

• Daily 15 minute status meeting;
• Same place and time every day;
• Meeting room;
• Chickens and pigs;
• Three questions;
  - What have you done since last meeting?
  - What will you do before next meeting?
  - What is in your way?
• Impediments; and
• Decisions
Chickens and Pigs

A chicken and a pig are together when the chicken says, "Let's start a restaurant!"

The pig thinks it over and says, "What would we call this restaurant?"

The chicken says, "Ham n' Eggs!"

The pig says, "No thanks. I'd be committed, but you'd only be involved!"
What Is Being Made Visible?

• When a Team member says “done,” what does that mean?

• Code adheres to standards, is clean, has been refactored, has been unit tested, has been checked in, has been built, and has had a suite of unit tests applied to it

• Development environment for this to happen requires source code library, coding standards, automated build facility, and unit test harness
Exercise:
The truly dysfunctional Daily Scrum
• 8 volunteers for the team
• 1 volunteer ScrumMaster
Sprint Review Meeting
What Is Being Made Visible?

• When the Team says “done,” what does that mean?

• Maintaining trust with customer by not “hiding” undone work.

• Functionality has been code reviewed, functionality has been integrated and built, acceptance tests have been run, and documentation has been created.

• QA environment for this has automated acceptance testing tools.
Evaluation Consequences:

1. Restoring unfinished functionality to the Product Backlog and prioritizing it.
2. Removing functionality from the Product Backlog that the team unexpectedly completed.
3. Working with the ScrumMaster to reformulate the team.
4. Reprioritizing the Product Backlog to take advantage of opportunities that the demonstrated functionality presents.
5. Ask for a release Sprint to implement the demonstrated functionality, alone or with increments from previous Sprints.
6. Choosing not to proceed further with the project and not authorizing another Sprint.
7. Requesting that the project progress be sped up by authorizing additional teams to work on the Product Backlog.
Sprint Retrospective

• Process improvement at end of every Sprint
• Facilitated by ScrumMaster
• What went well, what could be improved.
• ScrumMaster prioritizes based on team direction
• Team devises solution to most vexing problems

“Project Retrospectives,” Norman Kerth
Exercise MLBTix Part 2

At the Sprint Review on February 27, the Team has demonstrated to the Commissioner an almost ready to install product, scalable to the requirements and with all the functionality. The Commissioner shares good news with the team; 5x the tickets will be bought and sold and 40m users will sign on in the first 1 second.

1. Form teams of 5. One is the Team, the other the Commissioner.

2. Select someone from each team to play the role, using tag-wrestling tactics.

3. The “Team” and Commissioner carry on the resulting dialogue, with the “Team starting the conversation.
Plan the project

WHY plan??

• Lay out a common set of understandings from which emergence, adaptation and collaboration occur.

• Establish expectations that progress will be measured against.

• Convince a source of funding that the ROI of this project is worthwhile.
Ideal Team Day

One day of work of all the skills applied to build a piece of an increment.

Includes all parts of the increment, including documentation, testing, code, and other required artifacts.

Consists of all people with the skills to build such an increment.

One thread of effort.

Sustainable pace
The Basics

Ideal Team

1 programmer
1 tester
½ analyst/designer
½ technical writer

Average 2.5 ideal teams per development team

Sprint Size, 16 days/Sprint

Average 2.5 x 16 = 40 ideal team days/Sprint

Sprint: 1
Capacity: 40 ITD
Allocation:
Velocity

Number of product backlog requirements converted into potentially shippable functionality/$100,000 investment.

<table>
<thead>
<tr>
<th>Months since Scrum implemented</th>
<th>1</th>
<th>12</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Productivity</td>
<td>4.5</td>
<td>9.0</td>
<td>12.2</td>
</tr>
<tr>
<td>Quality</td>
<td>100+</td>
<td>100</td>
<td>5</td>
</tr>
</tbody>
</table>
Product Backlog

• List of functionality, technology, issues
• Issues are placeholders that are later defined as work
• Emergent, prioritized, estimated
• More detail on higher priority backlog
• One list for multiple teams
• Product Owner responsible for priority
• Anyone can contribute
• Maintained and posted visibly
• Derived from Business Plan or Vision Statement, which sometimes have to be created with customer
Backlog

Attributes:

Cost (ITD’s), Tests, Value ($, H/M/L, x/1000)

Relationships:

Sprint, Function, Specification, Task

Average Size: Varies across time but seems to average:

15 ITD’s for top 20%
20 ITD’s for next 20%
30+ ITD’s for last 60%

• 80% of the value comes from 20% of the functionality
• 60% of the functionality delivered in successful projects is rarely or never used.
Prioritize Product Backlog

1. Allocate 1,000 arbitrary units of value among the Product Backlog
2. Sort by Value
3. Spend $.4TD/feature costing top 20%
4. Spend $.2TD/feature costing next 20%
5. Spend <.1TD/feature costing rest
6. OR allocate your planning time 60% to top 20% in value, 30% to next 20%, and 10% to remainder
7. Create Calculated Priority from value/cost
8. Re-sort on Calculated Priority
9. Manually adjust positions of product backlog items to fine tune

Each feature averages 5/10 requirements, which have different values
Sprint: 1
Capacity: 40
Allocation: 42TD
<table>
<thead>
<tr>
<th>Drag</th>
<th># of years together</th>
<th>Knowledge of technology</th>
<th>Knowledge of domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>&lt; 3 months</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>0.75</td>
<td>&lt; 3 months</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>0.7</td>
<td>&lt; 3 months</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>0.75</td>
<td>&lt; 3 months</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>0.5</td>
<td>&lt; 3 months</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>0.5</td>
<td>&lt; 3 months</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>0.75</td>
<td>&lt; 3 months</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>0.5</td>
<td>&lt; 3 months</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>0.35</td>
<td>&lt; 3 months</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>0.6</td>
<td>&lt; 1 year</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>0.55</td>
<td>&lt; 1 year</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>0.5</td>
<td>&lt; 1 year</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>0.55</td>
<td>&lt; 1 year</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>0.3</td>
<td>&lt; 1 year</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>0.25</td>
<td>&lt; 1 year</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>0.5</td>
<td>&lt; 1 year</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>0.25</td>
<td>&lt; 1 year</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>0.2</td>
<td>&lt; 1 year</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>0.5</td>
<td>&gt; 1 year</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>0.45</td>
<td>&gt; 1 year</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>0.4</td>
<td>&gt; 1 year</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>0.45</td>
<td>&gt; 1 year</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>0.35</td>
<td>&gt; 1 year</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>0.2</td>
<td>&gt; 1 year</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>0.4</td>
<td>&gt; 1 year</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>0.2</td>
<td>&gt; 1 year</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>0</td>
<td>&gt; 1 year</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

**Adjusted estimate =**

\[
\text{Estimate} \times (1 + \text{drag factor})
\]

Copyright 2003-2005, ADM, All Rights Reserved v6.0
If teams are collocated,

Adjusted estimate =

Estimate * .6
Daily Scrums per Sprint

9:00AM
9:15AM
9:15AM
9:30AM
9:30AM
9:45AM
9:45AM
10:00AM

Sprint 1
Sprint 2
Sprint 3

Coordinating Scrum of Scrums

If more than one team,
Adjusted estimate =
Estimate * 1.4
## Release 5.0

<table>
<thead>
<tr>
<th>Sprint</th>
<th>Capacity</th>
<th>Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40TD</td>
<td>42TD</td>
</tr>
<tr>
<td>2</td>
<td>40TD</td>
<td>44TD</td>
</tr>
<tr>
<td>3</td>
<td>80TD</td>
<td>80TD</td>
</tr>
<tr>
<td>4</td>
<td>120TD</td>
<td>122TD</td>
</tr>
<tr>
<td>5</td>
<td>120TD</td>
<td>124TD</td>
</tr>
<tr>
<td>6</td>
<td>120TD</td>
<td>90TD</td>
</tr>
</tbody>
</table>

**Goal - Beat the competition**
Build Plan as Needed,
(remembering not to build unordered inventory)

1. Already funded, underway project - detail inventory for next several Sprints

2. Unfunded, new project with trust - detail inventory to level needed to estimate based on history.

3. Unfunded, new project without history - detail inventory to level where reasonable likelihood of meeting initial plan.

4. Unfunded new project with distrust - detail all inventory and build trust during project.
Fixed Price, Fixed Date

Or

Latest Date, Maximum Cost

Contract provisions:

1. Any requirement that hasn’t already been worked on can be swapped out for another of equal value;

2. Priority of requirements can be changed;

3. Customer may request additional releases at any time at prevailing time and material fees;

4. Customer may terminate contract early if value has been satisfied for 20% of remaining unbilled contract value
Stage Gate/Milestone Driven Development

Product Backlog

Requirements - System

Go/Nogo

Product Backlog
Exercise: Deadlines

You and the team have planned the project. It will be done three months after the customer needs it.

1. What are some options available to you?
2. What do you think is the best option?
3. Who is best positioned to choose the best option?
Exercise: Organizational Capacity

All teams are fully consumed on projects; However, a really important prospect turns up and you want to get their work done in parallel with existing work.

• What are some options for handling this prospect, satisfying them, and turning them into a long term customer?

• What if there are twenty of these prospects?
Product Backlog Refactoring

1. Re-evaluate calculated priority of top 20% of Product Backlog every Sprint and reprioritize accordingly;

2. Granularize and estimate probable Product Backlog for next Sprint;

3. Have team allocate 5% of their Sprint time for this activity, which should be compartmentalized to minimize interruption; and

4. Never allow the Product Owner to go into the Sprint Planning meeting with an inadequate Product Backlog.
Report Progress

• Planned product backlog and releases.

• Revised product backlog and releases.

• Complete analysis of any changes in backlog, priorities, estimates

• Analysis of productivity

• Progress toward release

• Actions to improve
Detailed Reporting

Compares Planned To Actual
Detailed Reporting

Displays progress made every milestone or Sprint on Vision models

Business Architecture – Leasing Operations

Key:
- May 1
- Jul 1
- Sep 15
- Phase II

✓ Completed This Phase
Speaking Their Language

If you flip the Product Backlog on its side and

• Make Sprint Planning and Reviews into milestones

• Create dependencies

You have a chart on M/S Project that people are comfortable with understanding

If you replace the Product Backlog with the Sprint Backlog as a Sprint begins, you are building work inventory just in time in response to orders.
Story

... of how a large bank would be done with the application in 5.5 weeks, causing the customer to sit up and start getting ready for the imminent implementation.
Subject:
Managing to optimize value
The single, wringable neck!
Customer Collaboration over Contract Negotiation
80% of value from 20% of functionality

Agile Collaboration -

business value =

\[ f(\text{cost, time, quality, functionality}) \]

Fixed Price Contracts -

cost = \[ f(\text{time, quality, functionality}) \]
Managing a Release

- Business determines when a release is needed, what functionality it must contain, and what is an acceptable level of quality and cost.

- Development determines how long it takes to build the release.
  - Development creates preliminary estimates
  - Development refines the estimates as priority increases
  - Development selects the product backlog for development, each Sprint.

- Business focuses on business value derived from the release.
Project Management Variables

Velocity of turning Product Backlog into increments of functionality

Product Backlog

Time

Other variables: Quality, Value

Copyright 2003-2005, ADM, All Rights Reserved v6.0
Exercise

A Product Owner and other stakeholders inspect the increment at the end of the Sprint. They are dissatisfied.

What are some of the potential reasons for their dissatisfaction? List them. Identify one and discuss what the Product Owner can do to remedy it during the next Sprint.
Managing Release Workload

Project slope of work remaining to determine probable release date

By ninth month, not enough productivity to hit desired release date in 20th month

Customer reduced expected functionality in release which raised the line for release date.
Managing Earned Business Value

Cumulative

Earned Business Value = Value planned and budgeted in TD’s - work expended in TD’s
Story

... of large customer of Primavera’s needing just a few more pieces of functionality before they could commit to a $14m purchase of release 5.0.
Subject:
Implementing Scrum - Advanced Topics

Topic - Scrum Implementation Templates
Scrum Implementation Template

Everything around a Scrum project that impedes its productivity and quality is an impediment.
During “Scrum Aptitude Tests” management has constructed preliminary lists of organizational impediments. They have then formed Sprint teams and delivered improvements, or “done” organizational changes, at Sprint Reviews. The CEO, CTO, CIO is the product owner.

Exercise:

If these impediments have been known to management, why have they been allowed to persist until Scrum was initiated?
Organizational impediments to optimized software construction and delivery are uncovered during the Sprints.
Organizational impediments to optimized software construction and delivery are uncovered during the Sprints.

Management’s job is to prioritize and systematically remove these impediments. This is very hard work.
Scrum Implementation Template

Planned Implementation Overview
Given the template, how do we explain Wildcard’s success starting 29 Scrum projects at once, or Bentley System’s success starting 18 Scrum projects at once?
Artifact Redefinition

If all artifacts and documentation required by this organization haven’t been fully defined and aren’t well known to the development team, the following work must be done prior to delivering too many increments:

“Define all documentation and artifacts that are part of each increment of product functionality”
Subject: Closing activities
**Scrum Compliance with CMM Software Framework**

<table>
<thead>
<tr>
<th>Level</th>
<th>Key Practice Area</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Requirements management</td>
<td>✓✓</td>
</tr>
<tr>
<td>2</td>
<td>Software project planning</td>
<td>✓✓</td>
</tr>
<tr>
<td>2</td>
<td>Software project tracking and oversight</td>
<td>✓✓</td>
</tr>
<tr>
<td>2</td>
<td>Software subcontract management</td>
<td>✓✓</td>
</tr>
<tr>
<td>2</td>
<td>Software quality assurance</td>
<td>✓✓</td>
</tr>
<tr>
<td>2</td>
<td>Software configuration management</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>Organization process focus</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>Organization process definition</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>Training program</td>
<td>✓✓</td>
</tr>
<tr>
<td>3</td>
<td>Integrated software management</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>Software product engineering</td>
<td>✓✓</td>
</tr>
<tr>
<td>3</td>
<td>Intergroup coordination</td>
<td>✓✓</td>
</tr>
<tr>
<td>3</td>
<td>Peer review</td>
<td>✓</td>
</tr>
</tbody>
</table>
Short History of eXtreme Programming

From: Kent Beck
To: Jeff Sutherland <jsutherland>
Reply: 70761.1216@compuserve.com
Date: Mon, 15 May 1995 18:01:15 -0400 (EDT)
Subj: HBR paper

Is there a good place to get reprints of the SCRUM paper from HBR? I've written patterns for something very similar and I want to make sure I steal as many ideas as possible.

Kent
## Scrum and XP Comparison

<table>
<thead>
<tr>
<th>Scrum</th>
<th>XP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Backlog of requirements - less granular</td>
<td>Stories of Specifications - more granular</td>
</tr>
<tr>
<td>30 day iteration required to complete increment (starts with analysis)</td>
<td>1-2 week iteration required to create software increment (no analysis, just design)</td>
</tr>
<tr>
<td>Estimates gradually get better as a matter of course</td>
<td>Effort is made to increase precision of estimates</td>
</tr>
<tr>
<td>Customer interrelates at ROI level</td>
<td>Customer interrelates at specification level</td>
</tr>
<tr>
<td>Implements in 2 days, then gradually improves everything</td>
<td>Implements in 6-8 months, depending on existing engineering practices</td>
</tr>
<tr>
<td>Management process that wraps any existing business processes and methodologies</td>
<td>Engineering process that has borrowed some wrapping management practices, but is at odds with many organizational practices</td>
</tr>
</tbody>
</table>
Uncertain and Anxious

1. Everyone is.

2. This course has taught you how to think and make your own decisions within the principles and goals of Agile.

3. Very few decisions are worse than none.

4. Inspect and adapt give you many chances to upgrade your decisions.

5. Get Scrum Buddies.

6. Use Yahoo egroup ScrumDevelopment.

7. Go forth and multiply.
Closing Activities

1. Scrum Buddies.

2. Fill out evaluation form and turn it in.

3. Sign and email one license to turn in; blank one is signed.

Recommended Readings

- User Stories Applied, Mike Cohn, Pearson Education, 2004
- Extreme Programming Explained, Kent Beck, Addison Wessley, 2000
- American Ground, William Langewiesche, North Point Press, 2002
- Industrial Dynamics, Jay W. Forrester, MIT Press, 1961
- Complexity and Management, Ralph D. Stacey, Routledge, 2000
- Project Retrospectives, Norman Kerth, Dorset House, 2001
- The Art of Focused Conversation, Brian Stanfield, New Society Publishers, 2000
- The Alphabet Versus the Goddess, Leonard Shlain, Viking, 1998
- The Dreams of Reason, Heinz Pagels, Simon and Schuster, 1988
- The Knowledge Creating Company, Nonaka and Takeuchi, Oxford University Press, 1995
- The Pragmatic Programmer, Hunt and Thomas, Addison Wesley, 2001
- The Capability Maturity Model, Paulk et al, Addison Wesley, 1994
- Agile & Iterative Development, Craig Larman, Pearson Education, 2004
Scrum Resources

• The Scrum Alliance:
  (www.scrumalliance.org)

• Control Chaos:
  (www.controlchaos.org)

• Agile Alliance:
  (www.agilealliance.org)