GOSEI ACADEMY

Certified ScrumMaster® Training

Ran Nyman (CST)

Exercise

GOSEI ACADEMY

Imagine that the training has ended
You are walking out and turn to the person next to you and say:
"I'm so glad I attended this training. I really met my reasons for attending."

What were those personal objectives this training met for you?

Who am I

GOSEI ACADEMY

Agenda

GOSEI ACADEMY

- Introduction
- Self Managing
- Agile and Lean Overview
- Scrum Overview
- Scrum Master
- Quality and Done
- Engineering practices
- Retrospective of Day 1
- Product Owner
- Product Backlog
- Planning in Scrum
- Planning Sprints
- Daily Scrum
- Sprint Review
- Continuous Improvement
- Development Team
- Scaling-Up Scrum
- Organization

Introduction

No answers just questions

Lots of exercises and discussions

Practicalities

- Parking lot
- Day length

Exercise

Role in organization

Scrum familiarity

Exercise

Who are we?

Change is Hard ?

There is nothing permanent except change.

- Heraclitus 535 BC - 475 BC

The philosophers have only interpreted the world, in various ways. The point, however, is to change it.

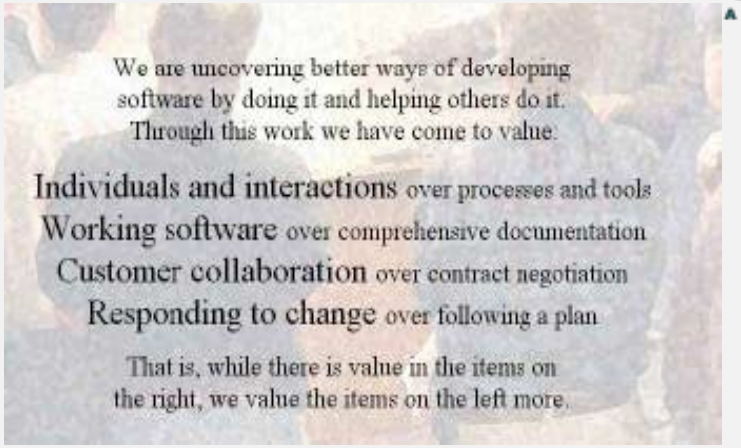
- Karl Marx

Software Development – Which Paradigm?

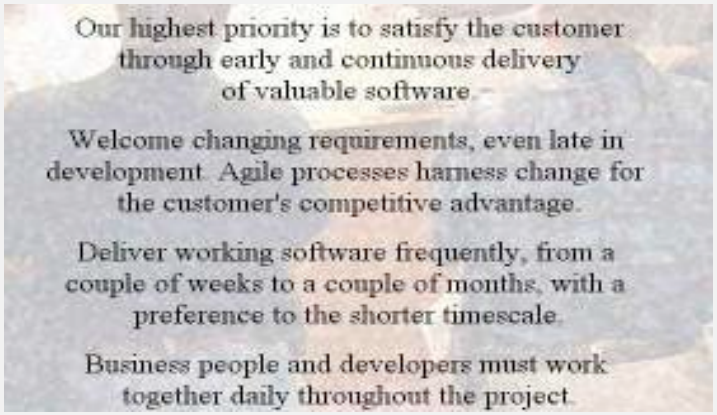
predictable; repeatable	large degree of uncertainty
process improvement	knowledge creation
low change rate	many changes
sequential steps	iterative; inspect & adapt
handoffs	concurrent and integrated
low communication	communication intensive
isolated specialization	collaborative specialization

History

Agile Manifesto



Agile Principles (1)



Agile Principles (2)

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Agile Principles (3)

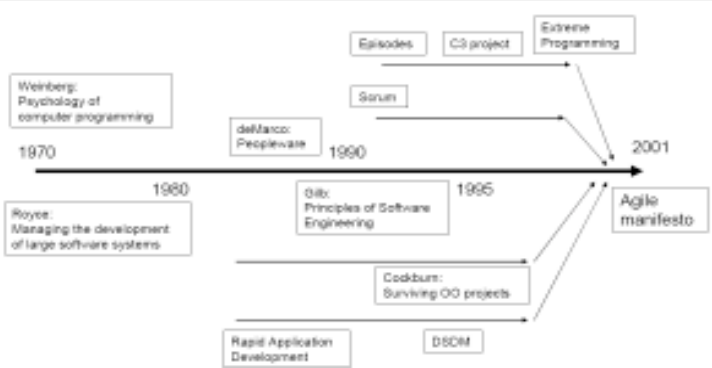
Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

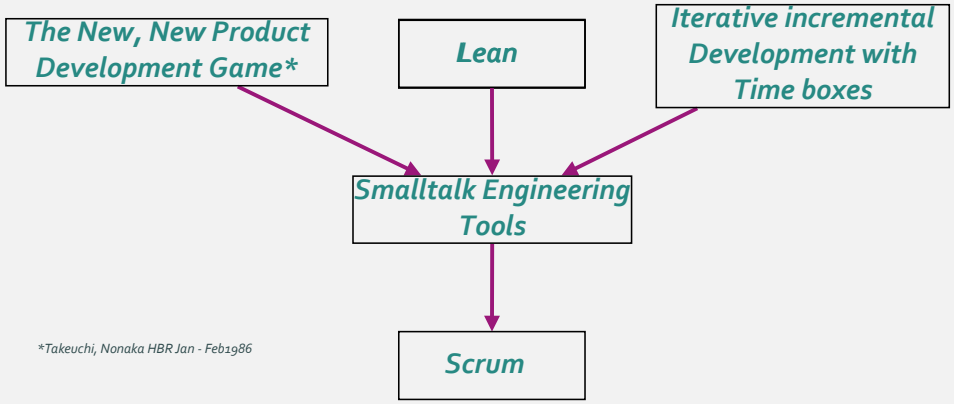
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

History of agile



Scrum

Where did it come?



*Takeuchi, Nonaka HBR Jan - Feb1986

Lean HBR Feb 2006 Hamel

"... it was only after American car makers had exhausted every other explanation for Toyota's success – an undervalued yen, a docile workforce, Japanese culture, superior automation - that they were finally able to admit that Toyota's real advantage was its ability to harness the intellect of 'ordinary' employees."

Exercise:

Discuss in your table groups what is the idea behind these quotes:

"The most dangerous kind of waste is the waste we do not recognize."

- Shigeo Shingo

"Where there is no Standard there can be no Kaizen (Improvement)"

- Taiichi Ohno

"Costs do not exist to be calculated. Costs exist to be reduced."

- Taiichi Ohno

The Secret Behind Lean Product Development

The goal of lean product development is learning

What is the goal in traditional product development?

Self-Management

Exercise command and control

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.

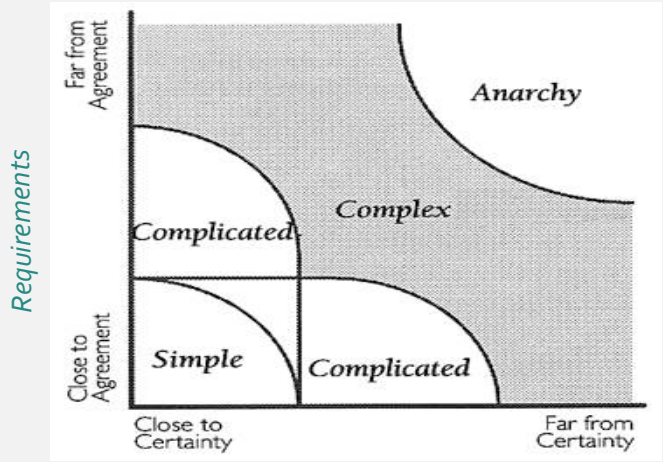
Exercise self-management

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.

Mission Command vs Detailed Command

Mission Command		Detailed Command
• Probabilistic • Unpredictable	Assumes war is	• Deterministic • Predictable
• Disorder • Uncertainty	Accepts	• Order • Certainty
• Decentralization • Spontaneity • Informality • Loose rein • Self-discipline • Initiative • Cooperation • Acceptable decisions faster • Ability all echelons • Higher tempo	Tends to lead to	• Centralization • Coercion • Formality • Tight rein • Imposed discipline • Obedience • Compliance • Optimal decisions, but later • Ability focused at the top
• Implicit • Vertical and horizontal • Interactive	Communication types used	• Explicit • Vertical • Linear
• Organic • Ad hoc	Organization types fostered	• Hierarchic • Bureaucratic
• Delegating • Transformational	Leadership styles encouraged	• Directing • Transactional
• Art of war • Conduct of operations	Appropriate to	• Science of war

Figure 1-4. Concepts of Command and Control



Traditional Management

Supervisors create organizations where employees are driven by management, not by customers, and where conformity becomes more important than creativity

Change from traditional manager to Scrum is very hard

- Old habits die hard
- Power is corruptive
- Different realities

What Makes Organization Effective?

There are many competing ideas what kind of organization is effective

Most based on beliefs and misunderstandings

Hards facts from:

- Relay assembly experiments. The most comprehensive study on manufacturing productivity 1927–1932
- Bank wiring room experiments 1931-1932

Social forces have biggest affect on productivity

Different Ways to Organize Real Work

Semco

Morningstar company

Gore

Dabbawala

Empirical and Defined Process

Predictive vs Empirical

Empirical Process

"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

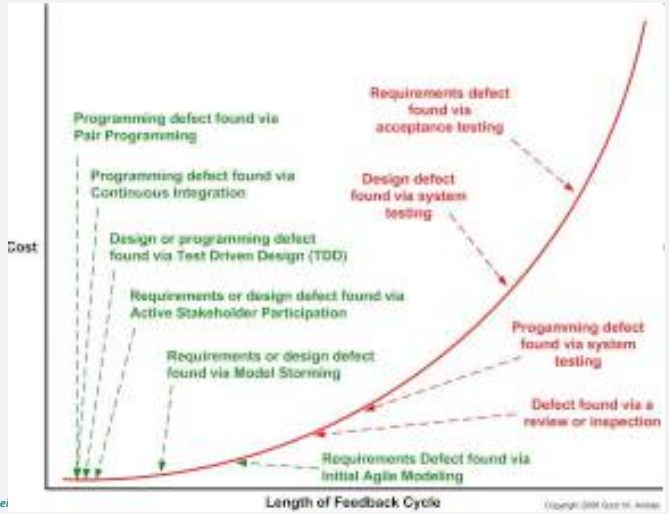
Empirical Process Control

Inspection and subsequent adaptations to optimize path for meeting goals

Transparency is required for inspection and adaption to work

Transparency requires courage and change how people are rewarded

Transparency ?



Compare traditional and agile methods

Little	*Amount of planning*	*A lot*
Little	*Predictability of execution*	*A lot*
Little	*Discipline required*	*A lot*
Little	*Documentation produced*	*A lot*

Scrum Overview

Scrum

Empirical process for managing the development and deployment of complex products.

Iterative, incremental, cyclic process

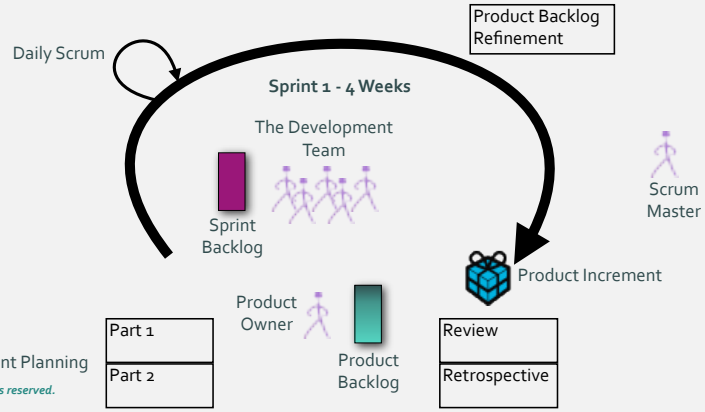
Time-boxed development

Evidence-based planning

Inspect and adapt - feedback loops

Cross-functional, self-managing teams

Scrum Overview



Product Owner

Owns the Product Backlog

Drives the development effort towards business needs

Responsible for

- Defining the features of the product
- Deciding on release date and content
- Prioritizing the work of the Development Team
- Being available and answering questions
- Making timely product decisions
- Accepting / rejecting work results

Development Team

Self-organizing

Cross-functional with no roles

Responsible for committing to work

Authority to do whatever is needed to meet commitment (inside Scrum Framework)

Owns the Sprint Backlog

Owns the development process

Develops working software to the best of its abilities

Scrum Master

Responsible for:

- Keeping the Team and process "well oiled"
- Facilitating key activities
- Providing information about useful techniques
- Removing barriers identified by the Development Team
- Shielding the Development Team from external interference
- Teaching the organization to use Scrum

Exercise

What feedback loops Scrum introduces?

Scrum Values

Commitment

Focus

Openness

Respect

Courage

Scrum Master

Scrum Master is NOT a Traditional Manager

Part of traditional manager's responsibility was to "supervise subordinates"

You just have to control subordinates. Supervisors can do this:

- Telling people what to do and then make sure they do it properly
- Maintaining the right to authorize decisions of subordinates
- Limit the information or resources available to them

Scrum Master Is NOT a project manager

Team manages it self

Coordinates its work with other people and teams

Organizes it's internal work

Product Owner makes decisions on content and schedule

No authority → Team makes decisions

Scrum Master

Challenges the organization; Is a change agent

Scrum master removes impediments when team asks

Facilitates scrum process

Remember dead scrum master is useless scrum master

Scrum Master

Remove barriers between development and customer so customer can drive development

Teach the product owner how to maximize ROI and meet objectives using scrum

Teach the organization how to apply scrum

Scrum Master Improves

Lives of the development team by facilitating creativity and empowerment

The productivity of development any way possible

Engineering practices and tools so each increment of functionality is potentially shippable

Scrum Master Part 2

Video Terry Tate

Exercise

Divide project management responsibilities between Scrum Roles

Exercise

Scrum Master being part of the development team?


Tools for Scrum Master

Listening

Talking and explaining

Questioning

Consequences

Listening

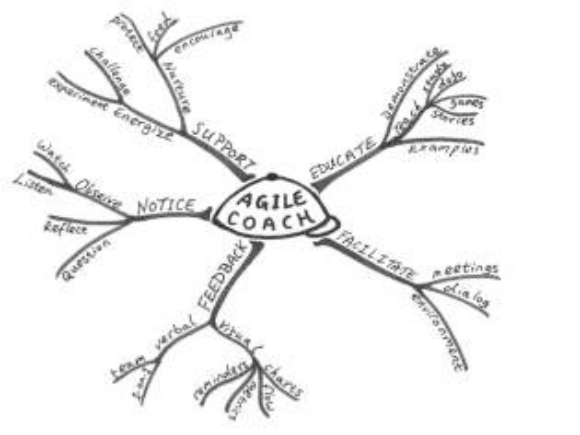
We first thought of presence as being fully conscious and aware in the present moment. Then we began to appreciate presence as deep listening, of being open beyond one's preconceptions and historical ways of making sense. - Peter Senge -

Active Listening

Active listening carries a strong element of personal risk. If we manage to accomplish what we are describing here — to sense deeply the feeling of another person, to understand the meaning his experiences have for him, to see the world as he sees it — we risk being changed ourselves … It takes a great deal of inner security and courage to be able to risk one's self in understanding another.

— Carl R. Rogers & Richard E. Farson

Scrum Master is an Agile Coach



From Davies & Sedley: Agile Coaching

Exercise

Practice listening

Exercise

What should you do when you notice that team decision could be wrong?

People Learn from Mistakes

Making mistake shows that you have tried something new

No mistakes means that you do not learn anything new

Fail fast ↔ Fail safe ↔ Safe fail

Controlled Failures

Team decision not right?

- Have to let them learn themselves
- Are you sure you are right?

Often people need to do it themselves before they learn

- Short cycles

Day of Scrum Master

How is my product owner doing?

How is my team doing?

How are engineering practices doing?

How my organization is doing?

Ask the Team

Example:

- I noticed <situation> what shall we do?
- I observed <situation> is that important?
- I feel <feeling> do you share that?
- Shall we try to find out why <situation>?
- What do you think we should do?
- Who has any idea about …?
- Is this useful?
- What have you decided?
- What should I do?

Exercise

Customer with urgent problem

MLBTix

MLBTix Exercise

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?

MLBTix (1)

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.

MLBTix (2)

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: March 30, 2004 – MLBtix site is up. Buyers and sellers can register.

Sellers can

1. 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.

MLBTix (3)

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.

MLBTix (4)

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:

- 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.
- The development team members all live within easy commuting distance of the development site.
- There are currently cubicles in the development site.
- The development environment is wireless and has all power and networking capabilities already operating.
- The development environment uses Microsoft development tools such as Visual Studio.
- 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.
- 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.
- All of the developers have excellent engineering skills, but they have only heard of Scrum and Extreme Programming, or used them sparsely so far.
- 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.
- All team members are baseball aficionados.
- A QA environment already exists.
- There are no adequate testing tools, continuous build tools, refactoring tools, and VSS is perhaps not adequate for the job.

Quality

Core Functionality

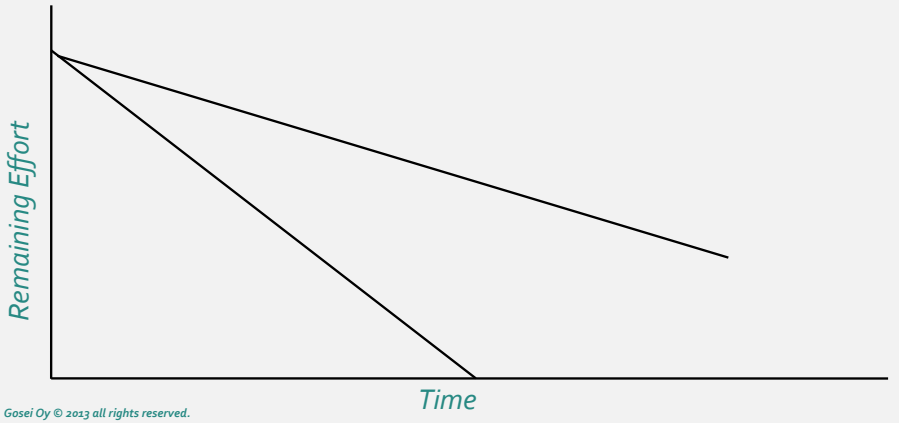
Most significant new functionality builds on it;

Also called infrastructure and legacy software;

Is fragile, doesn't have test harnesses, and few people still know how to or are willing to touch it; and,

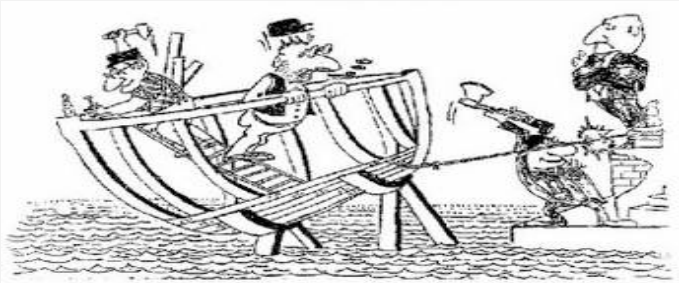
Requires more time to work on

Different Development Speed



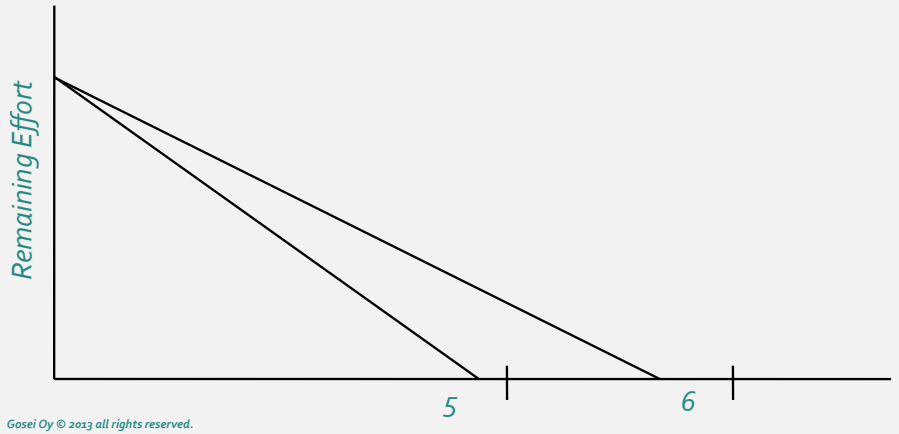
Where Does Core Functionality Come From?

Deadline

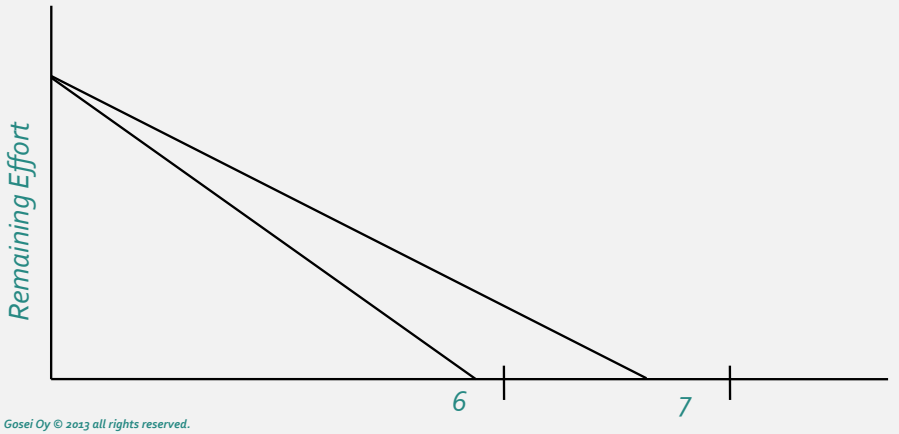


The Secret Toolbox

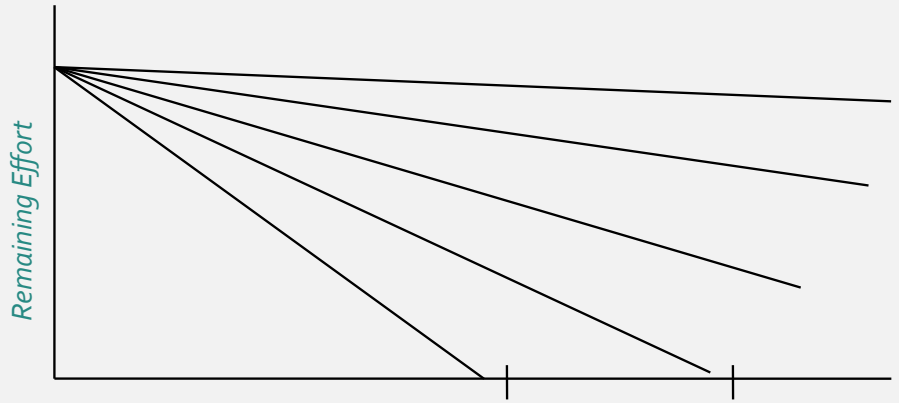
Where Does it Lead Us?



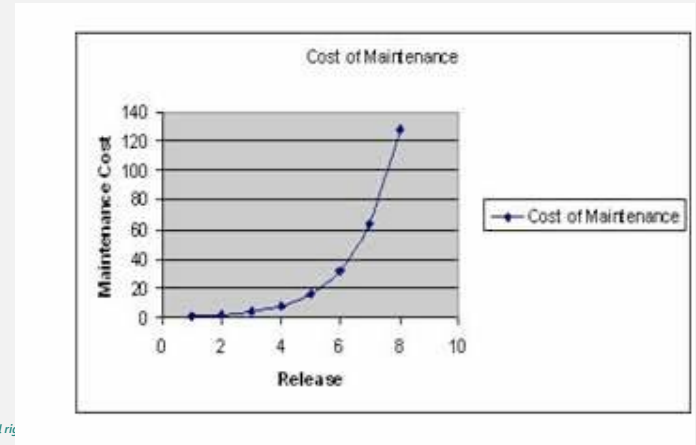
Where Does it Lead Us?



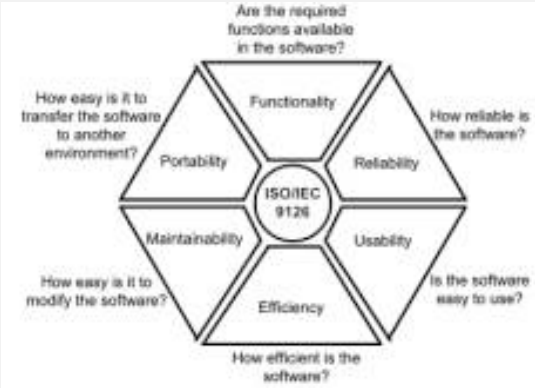
And Over Time



Design Dead SW



What is Quality?



Project Failures

"Equally responsible for the initiation of project with predefined failure is management that insists upon having fixed commitments from programming personnel prior to the latter's understanding what the commitment are for. Too frequently, management does not realize that in asking the staff for "the impossible", the staff will feel the obligation to respond out of respect, fear or misguided loyalty. Saying "no" to the boss frequently requires courage, political and psychological wisdom, and business maturity that comes with much experience."

Charles Lecht - "The Management of Computer Programming Projects" - 1967

Video Ken 13:36 – 15:54

Done

Exercise

GOSEI ACADEMY

What potentially shippable means?

Definition of DONE

GOSEI ACADEMY

Agreed with product owner and team

Reflects teams current capabilities of delivering potentially shippable product increment

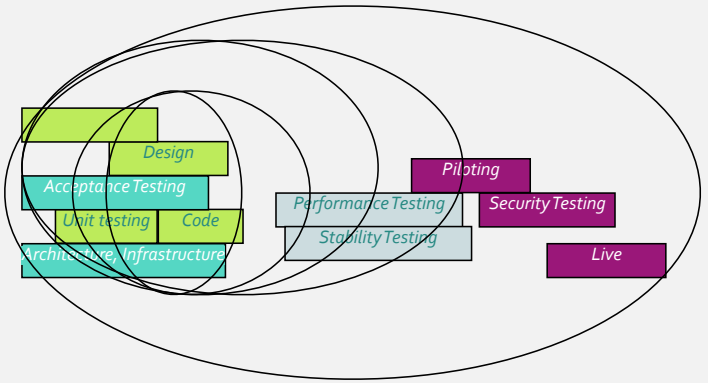
- Should be extended over time so everything that is needed to get product deployd
- Not DONE backlog items are not reviewed

Example definition of done

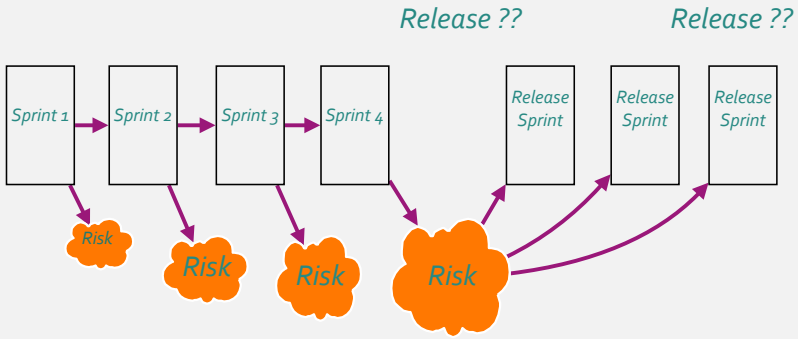
- Designed, coded, unit tested, acceptance tested, refactored, analyzed with code analysis tools, deployed

Extending Definition of DONE

GOSEI ACADEMY



UNDONE work



Exercise

Not DONE and still shown in sprint review?

Exercise

What are obstacles of extending definition of DONE?

Engineering Practices

Engineering Practices

Continuous integration (CI)

Collective code ownership

Acceptance test driven development (A-TDD) / Executable requirements

Slices of functionality and incremental development

Pair programming

Test driven development

Sustainable pace

Incremental Design

Incremental Design

Big Design Up Front → Requirement Change → More complexity, harder to change

Design Up Front → Requirement Change → Simpler design, adapts better to change

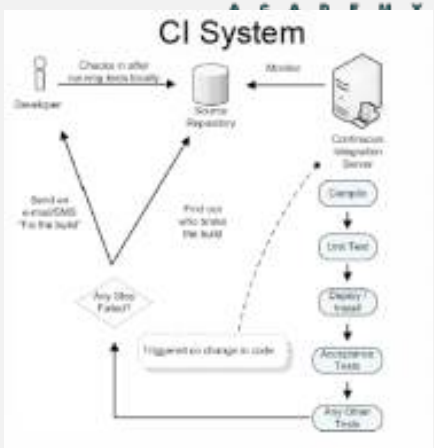
Continuous Integration

Practice of having always working software

Integration of code and test several times a day

Share what you have done to others

System that is building and testing



Tests as Specifications

Analysis creates tests; not speculations

Examples

- Drive out hidden assumptions
- Given when then
- Tabular format

Automated tests create visibility to progress

Running tested requirements

Test and Requirements

One of the most effective ways of testing requirements is with test cases very much like those for testing completed system.a

- Donald Gause & Gerald Weinberg, Exploring Requirements 1989

One of the most effective ways of specifying something is to describe in details how you would accept it if someone gave it to you.

- Bill Hezel The complete guide to software testing 1991

As formality increases tests and requirements become indistinquishable.

- Robert C. Martin & Grigori Melnik IEEE Software January/February Issue 2007

Exercise

Shared DBA

Exercise

What obstacles for introducing Scrum you can think of?

Scenarios

1) The VP of the group appears in the middle of the Sprint, and says to you: one of our clients has a special request which if we can complete it, we will win $1 million in business.

2) The product owner says that he's not going to be available to attend the Sprint planning activity, but he doesn't mind if the team goes ahead and does it without him.

3) In the middle of the Sprint, one of the team members manager comes and says she needs to pull him off the project for a couple days, to work on something else.

Scenarios (2)

4) One of the team members comes to you and tells you that the product owner just asked her to add an item to the current Sprint. Right now, the team is a third of the way through the Sprint.

5) The team appears to be very stressed out. They are having to work late most nights of the week, and they even have to work Saturdays every now and again, in order to meet their Sprint goals. You hear comments like scrum is awful it forces us to work so hard.

6) A team member has recently moved. He mentions that he is trying to find daycare for his daughter and it's blocking him from doing his work.

Retrospective

What should we improve for Day 2?

Exercise

99 Quality Balloons

Day 2

Exercise

Take a pair and try to remember what was important from yesterday.

Product Owner

New Role

Product Management was used to "throwing the project over the wall" and holding engineering / development responsible for meeting needs. Scrum puts this responsibility back on the Product Owner and customers through the inspect and adapt using the Sprint Review

Make decisions regarding ROI every Sprint end.

Product Owner

Should have a vision for the product.

Defines the features of the product, decides on release date and content.

Is responsible for the profitability of the product (ROI).

Prioritizes features according to market value.

Can change features and priority for each Sprint.

Accepts or rejects work results.

Product Owner

Image by http://www.flickr.com/photos/revolweb/

Chief Engineer

Takeshi Uchiyamada was born August 17, 1946. He graduated from Nagoya University with a degree in applied physics in March **1969** and joined Toyota Motor Corporation (TMC) in April the same year.

In January **1994,** Mr. Uchiyamada became project general manager of Vehicle Development Center 2. In January **1996,** he became **chief engineer** of that center, which developed the Prius—the world's first mass-produced gasoline-electric hybrid car.

Chairman of the Board

What Does It Mean In Practice

Works with the team

- Participates in Daily Scrum
- Works with the team to clarify requirements
- Participates in Product Backlog refinement
- Participates in planning activities
- Meets with stakeholders to clarify Product Backlog priorities
- Participates in portfolio planning

Product Owner's Dual View

Pictures from http://einesaite.blogspot.com/2010/08/mia.html

Exercise

How could Scrum Master and Development Team help Product Owner?

Exercise

1) The Product Owner does not attend the estimation and backlog refinement workshop

2) The Product Owner says: The software must be shipped on-time, on budget with all features implemented and high quality

3) The Product Owner has not prepared the product backlog for the sprint planning activity that is about to start.

4) The Product Owner is not available during the sprint to answer questions from team

5) The Product Owner says that certain features cannot be decomposed any smaller to make them fit into the sprint

Agile Planning

What Makes Planning Agile?

Is more focused on planning than the plan

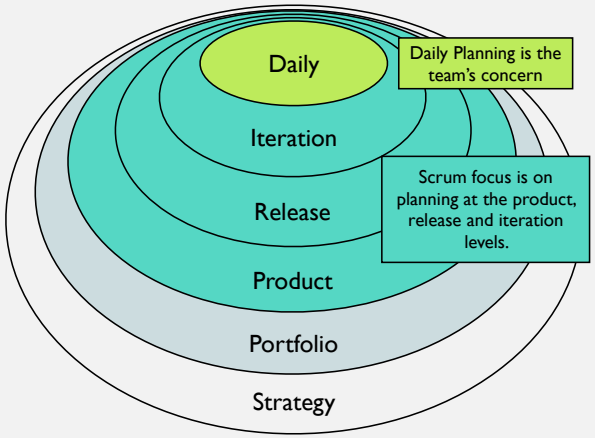
Encourages change

Result is plans that are easily changed

Is spread throughout the project

Balances effort with probability of change

Planning Levels



Product Vision

Vision

The product vision tells us how we are going to satisfy the customer needs

Value proposition and key characteristics of the software

A vision is necessary to:

Understand what functions the product will provide

Create focus and alignment

Prevent changing goals and requirements churn

Enabling a team to self-organize

Avoid "technical successes"

What the Vision should Answer?

Who is the customer?

What is the customer's problem?

How does the product solve the problem and add value?

What are the benefits compared to other solutions that are available to the customer?

On what basis will the customer judge the software?

Establish A Vision

Some tools:

- Elevator Statement
- Product Boxing
- Future Press Release

Elevator Statement

Explain the fundamentals in the time it takes to ride 15 floors

Syntax focuses the statement:

For (target customer)

Who (statement of the need or opportunity)

The (product name) is a (product category)

That (key benefit, compelling reason to buy)

Unlike (primary competitive alternative)

Our product (statement of primary differentiation)

Elevator Statement - Example

For [frequent travelers]

Who [want an all in one travel service]

[Travel Mate] is a [virtual travel agent]

That [will do everything for my trip]

Unlike [Itinerary Planner]

Our Product [will remove the need to visit any other website or shop for a trip]

Product Boxing

Design the vision of a box for your product (even if it won't come in a box)

3-4 bullet points

Future Press Release

What would you like the press release to read at the end of the project?

What are they key points you'd like pointed out?

What quotes would you have and who would they be from?

Press Release Template

_____ announced today the successful completion of the _____ project. This project provides _____ _____.
The customer for this project, _____, indicated in a recent interview that they selected _____ as their supplier due to the following key benefits:

1._____
2._____
3._____

_____ also identified several features that they thought were particularly useful. These include:

1._____
2._____
3._____

_____ noted that the single most important benefit of this project was _____ _____ _____"

Product Backlog

Product Backlog

Product Backlog												
Backlog item	Wiki link	Origin	Value	0 654	1 614	2 614	3 614	4 614	5 614	6 614	7 614	8 614
Item #1	http://wiki/item	Customer 1	1000	8	0	0	0	0	0	0	0	0
Item #2	http://wiki/item	Customer 2	500	8	0	0	0	0	0	0	0	0
Item #3	http://wiki/item	Improve	10	4	0	0	0	0	0	0	0	0
Sprint #1				20								
Item #4	http://wiki/item	Customer 1	2000	4	4	4	4	4	4	4	4	4
Item #5	http://wiki/item	Customer 5	100	2	2	2	2	2	2	2	2	2
Item #6	http://wiki/item	Customer 3	300	3	3	3	3	3	3	3	3	3
Item #7	http://wiki/item	Customer 3	200	13	13	13	13	13	13	13	13	13
Item #8	http://wiki/item	Customer 2	500	13	13	13	13	13	13	13	13	13
Item #9	http://wiki/item	Customer 1	200	21	21	21	21	21	21	21	21	21
Item #10	http://wiki/item	Customer 1	100	8	8	8	8	8	8	8	8	8
Item #11	http://wiki/item	Customer 1	400	50	50	50	50	50	50	50	50	50
Item #12	http://wiki/item	Customer 1	200	60	60	60	60	60	60	60	60	60
Item #13	http://wiki/item	Customer 3	400	50	50	50	50	50	50	50	50	50
Item #14	http://wiki/item	Customer 7	200	60	60	60	60	60	60	60	60	60
:	-											
:	-											
Item #30	http://wiki/item	Improve	400	50	50	50	50	50	50	50	50	50
Item #31	http://wiki/item	Customer 2	200	60	60	60	60	60	60	60	60	60
Release #3												
Item #32	http://wiki/item	Customer 1	400	50	50	50	50	50	50	50	50	50
Item #33	http://wiki/item	Customer 4	200	60	60	60	60	60	60	60	60	60
Item #34	http://wiki/item	Customer 1	400	50	50	50	50	50	50	50	50	50
Item #35	http://wiki/item	Customer 1	200	60	60	60	60	60	60	60	60	60

Product Backlog

Owned by the Product Owner

Anyone can contribute

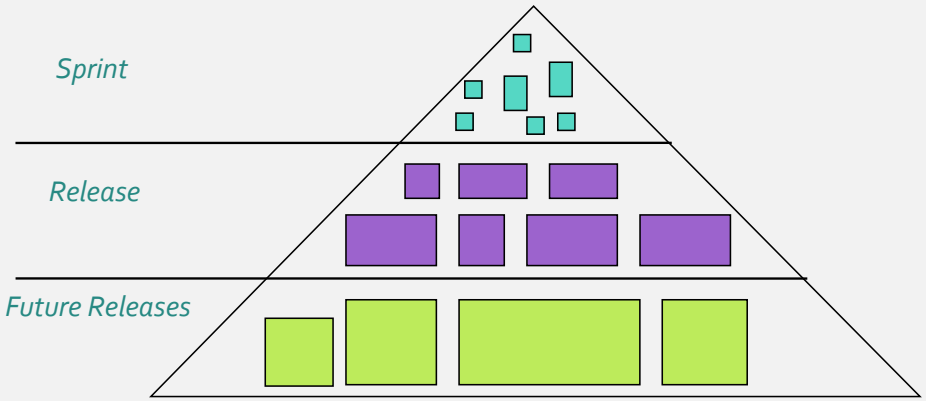
Derived from business plan and product vision

List of desired functionality and issues to address

Backlog items:

- Are ordered
- Have estimates
- Can have value and risk
- Are understood by the Product Owner
- More detail about highest priority items, less detail about lower priority items

Planning Levels



Possible PB information

Risk

Tests

Dependency to other product

Time the dependency is delivered

Person who knows best about it

Remember to keep it simple!

Updating Backlog

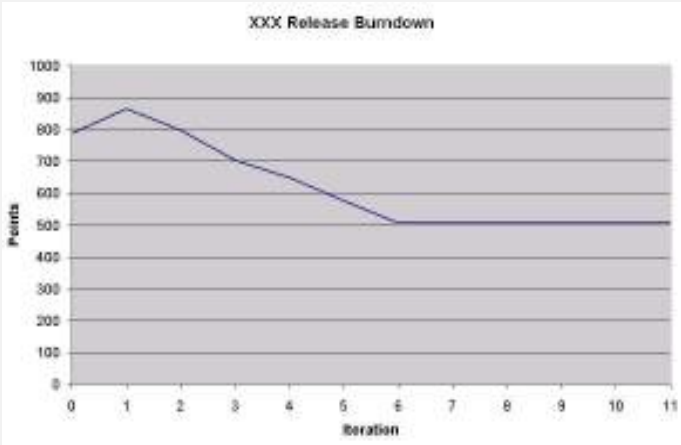
After each sprint product owner updates Product Backlog

- According to what team was able to get DONE during sprint

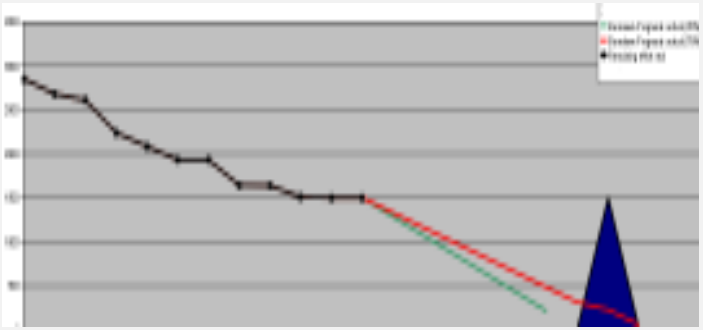
Items or either DONE or not DONE

- Partially completed items are not updates

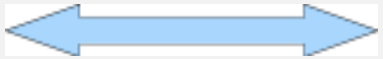
Release Burndown



Release Burndown 2



Product Backlog Items


Product Owner *Continuous conversation* *The Team*

PBL items are in problem domain

PBL items should have meaning to customer

Sprint backlog items are in implementation domain

Task that convert PBL items to potentially shippable product increment

User Stories

Card, Conversation and Confirmation

Example template:

- As a ... (user of the system)
- I want ... (feature or problem to be solved)
- So that ... (benefit of story being completed)

User Roles

Systems can have different users that

- are using the system differently
- have different needs and goals
- have different level of expertise

Identification of these roles help understanding the needs and expectations when developing features

Can help prioritization

Splitting Backlog Items

Stubbing out external dependencies

Splitting across supported data

Different configuration options

Splitting on operations performed

Different protocol messages

Splitting on CRUD operations

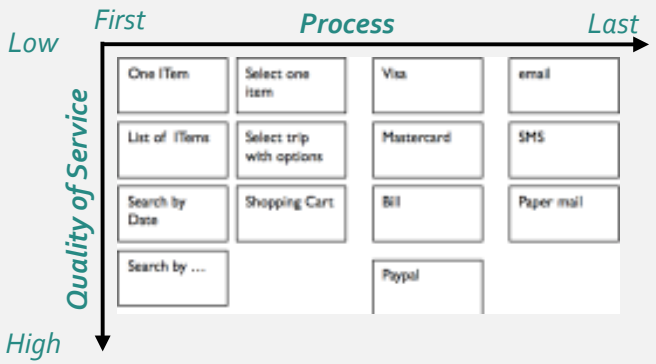
Removing cross-cutting concerns

E.g. security, logging

Splitting functional and non-functional

User Story Slicing

User Story Slicing Example



Splitting User Stories An Other Sample

As an operator, I want to provide access using PPPoE/PPP and EtherNet/DHCP

Two Obvious Parts

As an operator, I want to provide access using PPPoE/PPP (split the operations or services)

As an operator, I want to provide access using EtherNet/DHCP (split the operations or services)

As an operator, I want to provide access using EtherNet/ DHCP (split the operations or services)

As a user device (of a subscriber), I want to discover a DHCP server (split the operations - step 1 in "providing access using..."; also, move to the end-user view)

- As a user device, I want to discover a DHCP server using a fake (e.g., an alternative implementation of a c++ interface) DHCP service (split with a stub or io pathway)

As a user device, I want to discover a DHCP server using a fake (e.g., an alternative implementation of a c++ interface) DHCP service (split with a stub or io pathway)

- As a user device, I want to discover a DHCP server using a fake DHCP service with a hardcoded MAC address (split with data variation)
- As a user device, I want to discover a DHCP server using a fake DHCP service with a hardcoded MAC address on the ethernet pathway (split on io pathway)
- As a user device, I want to discover a DHCP server using a fake/stubs DHCP service with a hardcoded MAC address on the ethernet pathway underneath VLAN
- As a user device, I want to discover a DHCP server using real internal (within the gateway product) DHCP service (split on real/fake and io pathway)
- As a user device, I want to discover a DHCP server using real external DHCP service (split on real/fake and io pathway)

Product Backlog Refinement

Allocate 5-10% of every Sprint for working with the Product Owner, which could be compartmentalized to minimize interruption

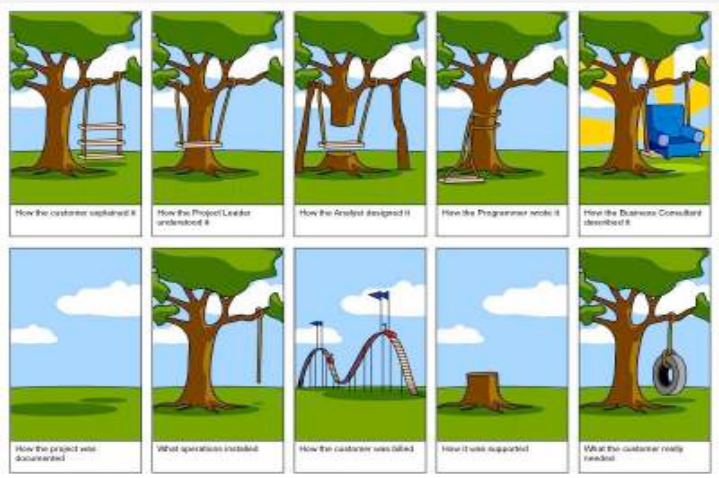
Work on the Product Backlog for next Sprints:

- Split , clarify and estimate probable items

Never allow the Product Owner to go into the Sprint Planning with an inadequate Product Backlog

(Also known as Backlog Refactoring, Backlog Maintenance, Backlog Grooming)

Initial Product Backlog



Exercise Initial Requirements (15 min)

Look at requirements for MLBTix.

Select out on what is needed for the March 30 release

Review functional and non-functional items.

Create a user story card for each Product Backlog item. There will probably be 10-15 cards.

NOTE: Do not strive for perfection, just the best you can do in 15 minutes

Exercise Assign Value (10 min)

Assign value to product backlog item

Value should reflect the relative value of each item on market

Non-functional items have value otherwise the system will be unusable

Distribute 1000 value points to items

Relative size estimates

Estimate Release Duration (15 min)

Everyone at table takes 8 cards 1,2,3,5,8,13,20 and BIG.

Choose a clear "about 4-5 day" item and assign it the value "3."

Choose a clear "about 2-3 week" item and assign it the value "8."

Play planning poker, one by one, vote on the relative size of the effort to "do" the product backlog items for the first release.

Discuss it further if large discrepancies in voting occur.

If close, use the higher number

Planning Poker

Wideband Delphi method

Fibonacci sequence

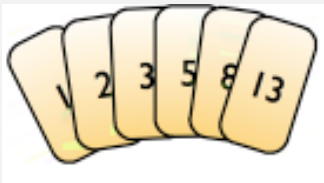
Clarification on the difference

Everybody involved

Quick

Reliable

Fun

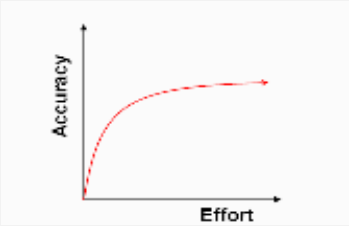


How much effort?

An "estimate" –it's just an educated guess

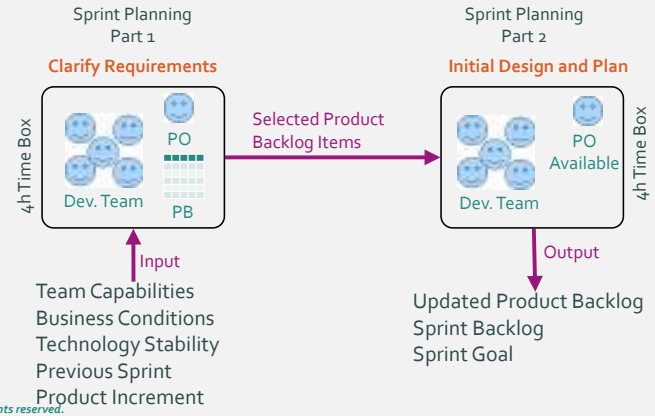
A little effort helps a lot

A lot of effort only helps a little more



Sprint Planning and Sprint Backlog

Sprint Planning



Doing Sprint Planning

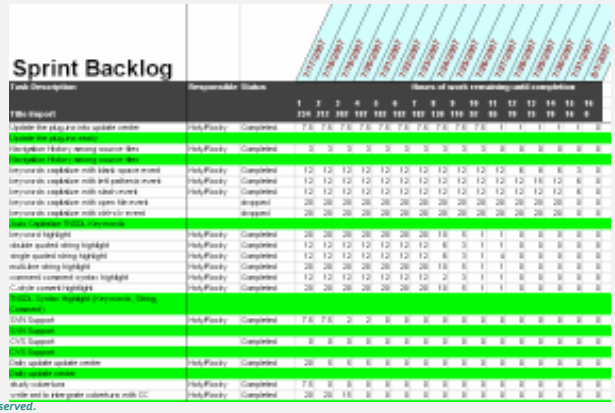
Sprint planning part 1

- Clarification on backlog item
- Tentatively select backlog items for Sprint

Sprint planning part 2

- Creating plan for coming Sprint
  - Don't forget backlog refinement
  - Maintenance?
- Commit to the items within development team
- Forecast to PO what get's Done

Sprint Backlog



Sprint Backlog

Plan to turn product backlog into working product functionality

Team members sign up for tasks, they aren't assigned

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 larger item … break it down later.

Update plan when more is known, as items are worked on

Sprint Backlog Estimates

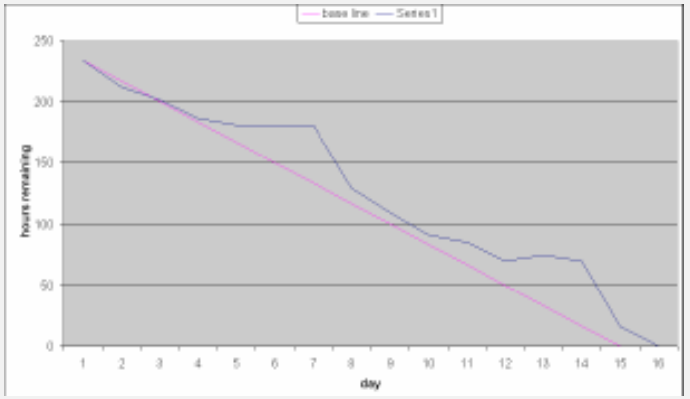
Work remaining reporting during a Sprint updates the estimated time required to complete planned items

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

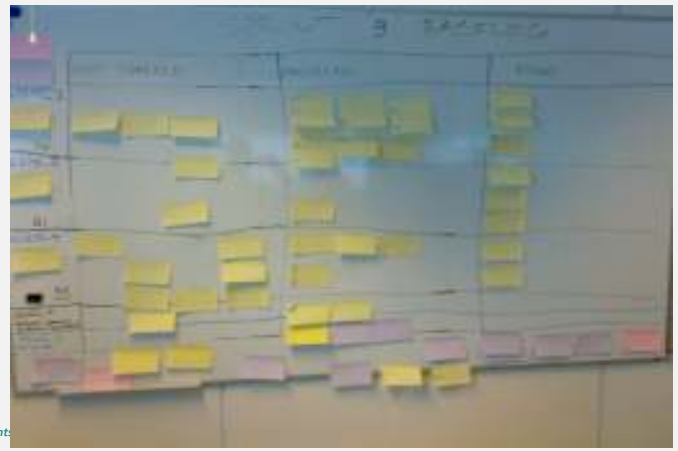
There are no mechanisms in Scrum for tracking the amount of time that a team works.

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

Sprint Burndown



Task Board



Sprint Burndown Chart

Exercise

Team refuses to update the sprint backlog

Daily Scrum

Daily Scrum

Daily 15 minute status and co-ordination activity

Same place and time every day

- Meeting / team room

Three questions

- What have you done since last Daily Scrum?
- What will you do before next Daily Scrum?
- What is in your way?

Impediments and Decisions

Usefulness

Keep asking yourself:

- "Does the team find the daily scrum useful?"

If the answer is NO, find out why!

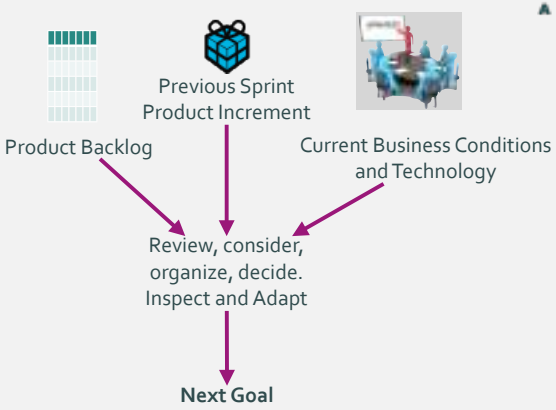
- Do they manage themselves?
- Do they share their work?
- Do they report unclear?
- Does it take too long?
- etc

Exercise

Daily scrum from hell

Sprint review and retrospective

Sprint Review



Retrospective

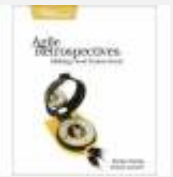
Process improvement at end of every Sprint

Facilitated by Scrum Master

What went well, what could be improved.

Team devises solution to most vexing problems

"Project Retrospectives," Norman Kerth



Exercise

Continuous Improvement

Retrospective

Gather information what happened

Analyze information

- Root cause analysis
  - Fish-bone
  - 5 Whys

Generate actions only 2-3 per sprint

Focus first on improvements that you / team can do

Time Line

Outcome of Retrospectives

Actions

Do we only have a few actions?

Are they considered useful?

Are they implemented?

Is "Done" extended?

Do we need to update our working agreements?

Do we need:

Tasks in the Sprint backlog?

Items in the Product backlog?

=> Improvement

Analysis and Action Plan

Exercise

Choose a real problem to solve.

Define the problem statement (What, Who, When, Where, Why?)

Draw the fishbone diagram and select categories (e.g. Process, People, Tools, Environment, Project)

Brainstorm factors in each category:

"What are the [category] issues causing or affecting the [problem]?"

Continue asking "Why is this happening?"

Stop when the causes are out side the team's control

Look for items that appear in more than one category

These are the most likely causes

Or discuss and dot-vote to find the most important causes

Decide what to do

- Long term goal
- Short term action in the next sprint

The Development Team

Development Team

Self-organizing

Cross-functional with no roles

Authority to do whatever is needed to meet commitment (with-in Scrum framework)

Open, collocated space

Resolution of conflicts

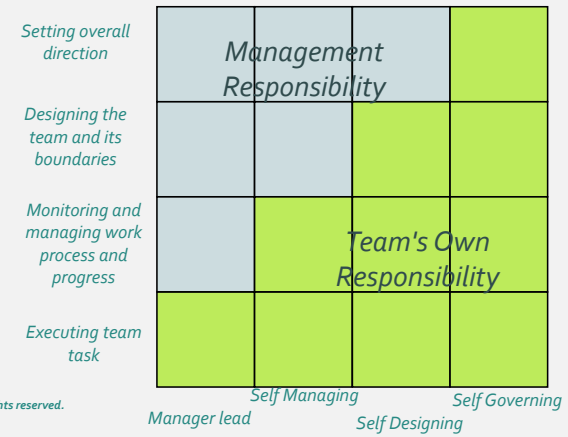
Self Managing Team

A group of employees who

- Have day-to-day responsibility for managing themselves
- Work they do with a minimum of direct supervision

- Members of self-managing teams typically:
  - Handle job assignments
  - Plan and schedule work
  - Make production and/or service-related decisions
  - Take actions on problems

Team Self-Management



Exercise

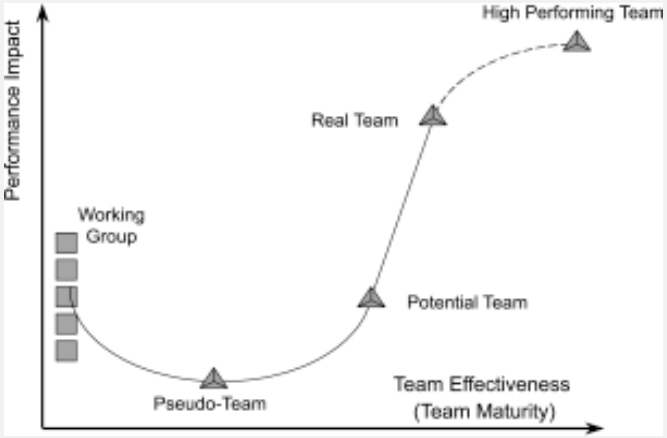
Angry analyst

Exercise

Learning to work together

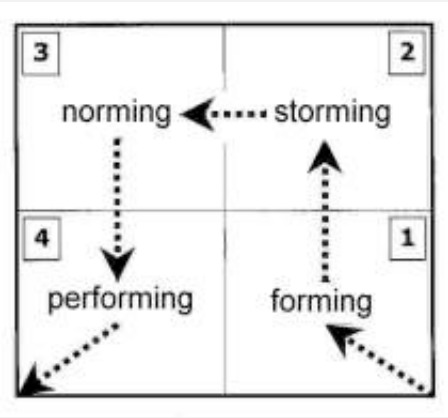
Five Dysfunctions of a Team

Team Performance



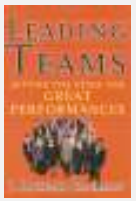
The wisdom of teams
Katzenbach, Jon R; Smith, Douglas K

Team Forming



Bruce Tuckman's 1965 Forming Storming Norming Performing Team-Development Model

Leading Teams



Working Agreements

How do we make decisions?

What is team name?

When is the daily scrum?

What happens if some one is late from daily scrum?

What engineering practices we use?

Are these good working agreements?

Decisions are made through discussion.

Write better quality code.

Document changes.

Notify if you cannot make it to Daily Scrum.

Work more with Product Owner.

Do whatever we agree to do in the time frame agreed.

Share leadership roles as and when appropriate.

Acknowledge and/or respond to all relevant communication within 1 day.

Set up a continuous integration machine.

Give constructive timely feedback to relevant person before complaining to someone else.

Are these good working agreements? (2)

Keep version controlled files in good shape

Write demo script for a user story before starting to implement it.

We will not work on user stories or items outside of the iteration scope.

Don't talk to phone in the team room.

Have frequent code reviews.

Write unit tests for implementation.

Let's start Daily Scrum on Time.

Have a code review every Thursday.

Everybody participates in planning activities.

Keep merges small.

Commit often.

Organization

Exercise

What changes Scrum introduces in organization?

Exercise

What makes Scum effective way to introduce change in organization?

Exercise

David Part 1

Exercise

David Part 2

Roles will Change

Scrum will be conflicting with lots of traditional organization roles and responsibilities:

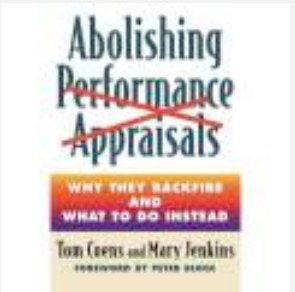
- Project manager will disappear?
- Line manager role will change.
  - Impediment backlog

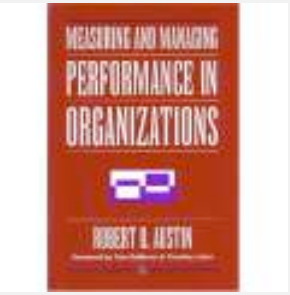
Other roles will change.

Always remember, this is a personal change in some persons future and career!

- Change is difficult and sometimes painful.

Performance Management





Start Improvement from Yourself

You need to gain the trust of the organization before they will be changing

Therefore, always focus on changing and improving yourself instead of blaming the organization!

Why would the organization believe any team that is not: Delivering working software without bugs every month!

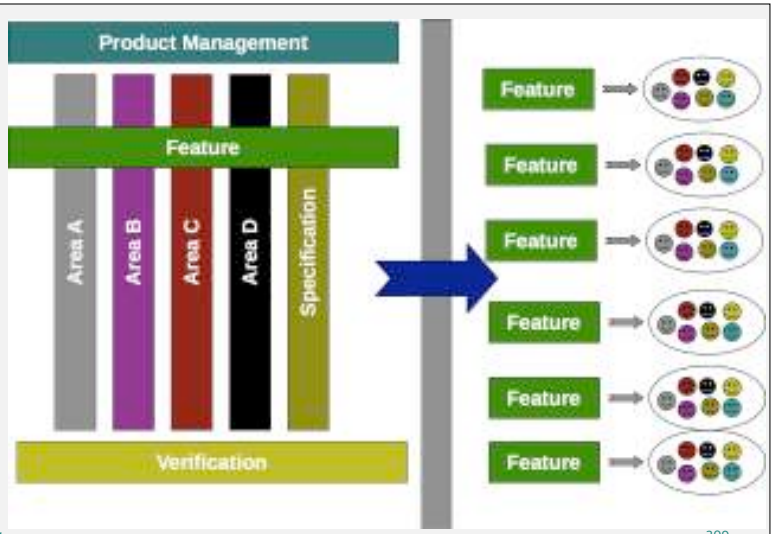
Impediments

Organizational impediments to optimized software construction and delivery are uncovered during the Sprints

Management's job it to prioritize and systematically remove these impediments

This is very hard work

Case



Exercise

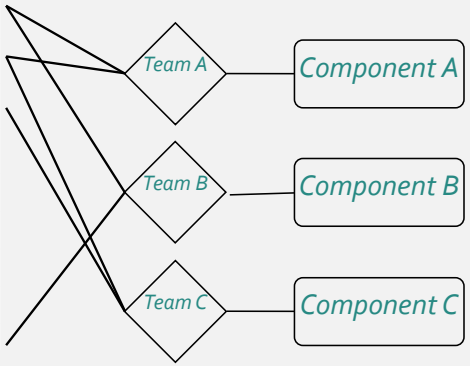
What would be on impediment backlog?

Scaling-Up

The more people you add the slower it will become so scaling-up is the last thing that you want to do

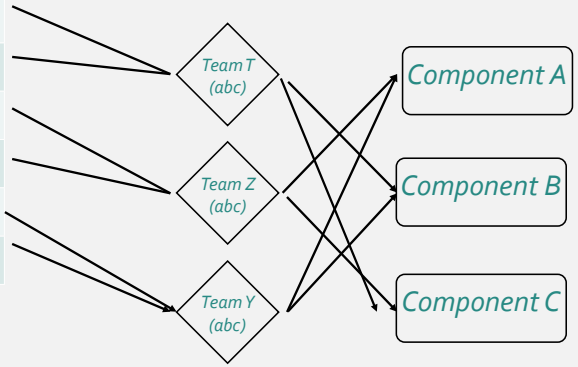
Component Teams

PBL Item	Team
Customer Centric Req	A, B
Customer Centric Req	A, C
Customer Centric Req	A, C
Customer Centric Req	A, C
Customer Centric Req	A, B, C



Feature Teams

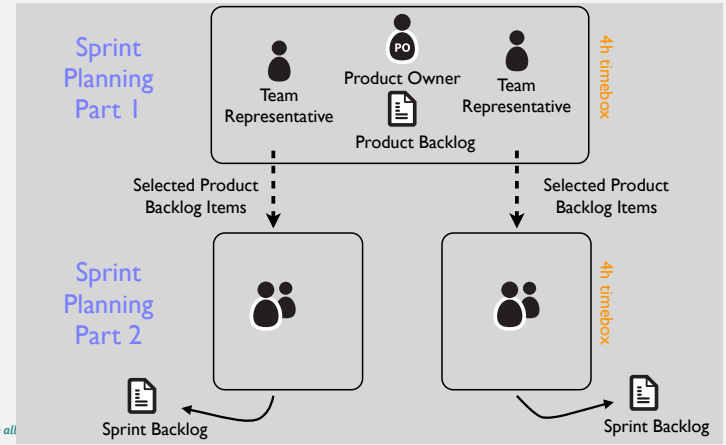
PBL Item	Comp
Customer Centric Req	A, B
Customer Centric Req	A, C
Customer Centric Req	A, C
Customer Centric Req	A, C
Customer Centric Req	A, B, C
Customer Centric Req	A



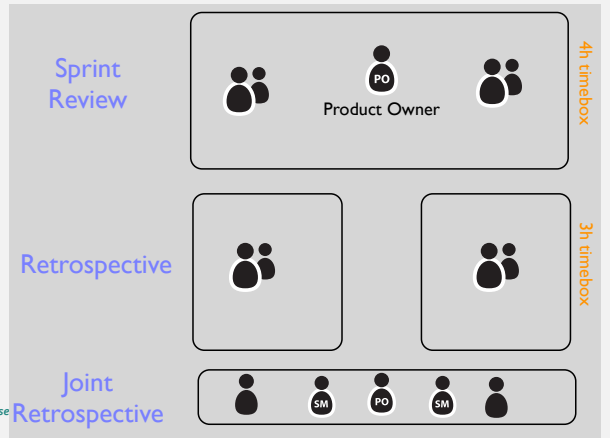
Feature Teams	Component Teams
optimizes to deliver customer value	optimizes to deliver maximum lines of code
focuses on high-value features	focuses on individual productivity and sacrifices high value features
completes customer centric requirements	completes only partials of customer features
avoids Conway's law	follows Conway's law
leads to customer focus	leads to 'invented' work
minimizes dependencies between teams which enable flexibility	dependencies between teams needs extra planning and leads to stiffness
focuses on learning and multiple specialization	focuses on single specialization
shares product code ownership	focuses on individual/team code ownership
shares responsibilities inside team	focuses on individual responsibilities
supports iterative development	leads to 'waterfall' development
requires skilled engineers	works with sloppy engineers
provides motivation to make changeable code	makes code rigid
looks difficult to implements	looks easy to implement

From Practices for Scaling Lean & Agile Development by Crag Larman and Bas Vodde

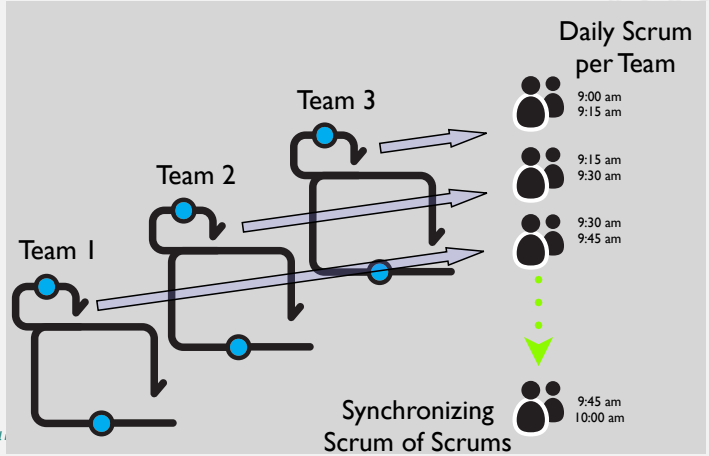
Scaling Sprint Planning



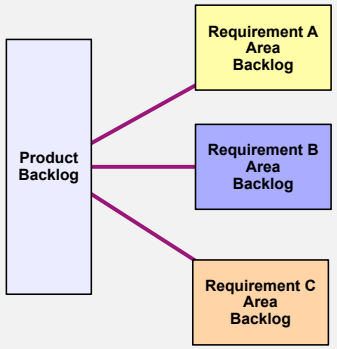
Scaling Sprint Review and Retrospective



Team Synchronization



Scaling Up Planning



Exercise

Create a media product using Scrum that communicates Scrum ideas (e.g., roles, artifacts, events, values/themes)

- Game, brochure, t-shirt, poem …
- Timing
  - Product vision 15 min
  - Initial product backlog 10 min
  - First Sprint 10 min
  - Second Sprint 10 min
  - Release

Closing

Resources

The Scrum Alliance:

- www.scrumalliance.org
- www.scrumalliance.org/why-scrum/resources

Agile Atlas

- www.agileatlas.org

Scrum list:

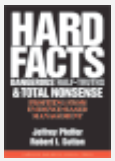
- http://groups.google.com/group/scrumalliance

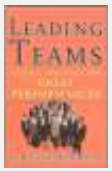
Agile Alliance:

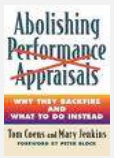
- www.agilealliance.org

Scrum Books

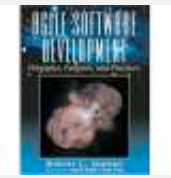

Management Books

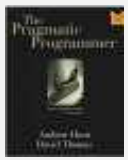





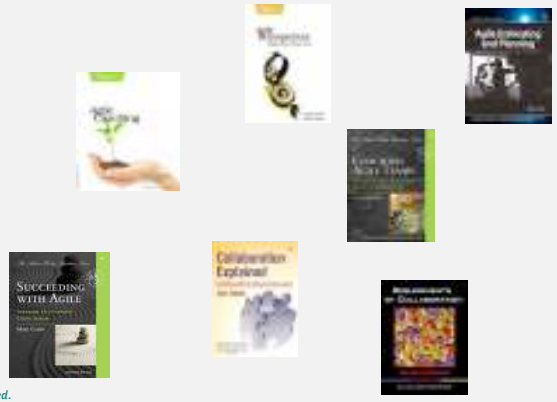


Technical Books

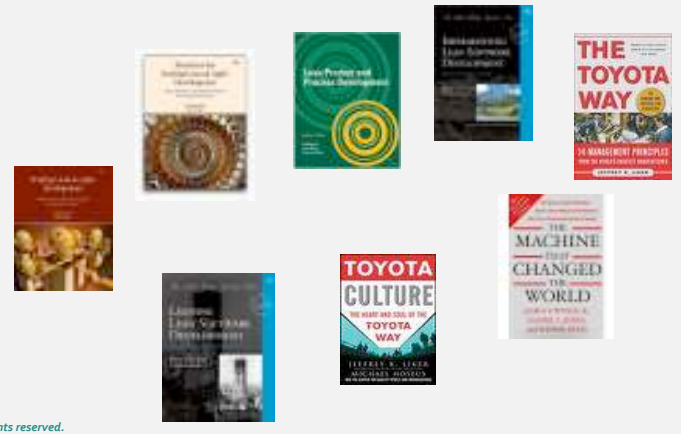





Agile Books



Lean & Agile Books



Q&A

Scrum Buddies

Extra Material

Sprint Abnormal Termination

Sprints can be canceled before the allotted time is 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 activity, where the reason for the termination is reviewed

Initiating Scrum Project

MetaScrum is responsible for staffing projects to maximize productivity and crossfunctional capabilities, minimizing centers of excellence, and maximizing team focus.

Fixed Price and Fixed Date or Latest Date and 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

Good Scrum Smells

Estimates are updated every day

Everybody is there at scrum on time every day

People offer to help others

People ask for help

People present the team with problems and solve them as a team

There's lots of talking and interaction

Lots of silly bits introduced by the team

Original slide:
Alan Atlas, Amazon

Bad Scrum Smells

The Sprint requires a lot more work than was planned

Team member reports the same item more than two days with the same or greater estimates and nobody notices or cares

No interaction outside of daily scrum

Product Owner not available for consultation

Distractions from outside the team

Hidden or multiple backlogs

Acceptance of the status quo

Failure to produce potentially shippable software every sprint

Original slide:
Alan Atlas, Amazon

Dispersed Team

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

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 are unproductive