COLLABNET

Certified Scrum Product Owner Course

Workbook

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

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.

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.

agile

Name: ______________________________

COURSE AGENDA

This course is not organized traditionally. During the first afternoon, you, the course participants, will collaboratively prioritize the possible course topics and select the topics that are most highly valued on this particular course.

However, some topics are ”musts”, because we will have to cover those in order for the course to be a certified ScrumMaster course. These ”must” topics are:

- **Agile Thinking** – *Agile values, 12 principles, predictable vs. Complex systems*
- **Scrum Framework** – *Meetings, roles, artifacts, self-organization, emergence*
- **Product Vision** – *Importance of vision to product success, testing templates*
- **Release Planning** – *Basic techniques for creating longer term plans*
- **Sprint Planning and Execution** – *Sprint meetings in practice*

The optional topics to prioritize from are:

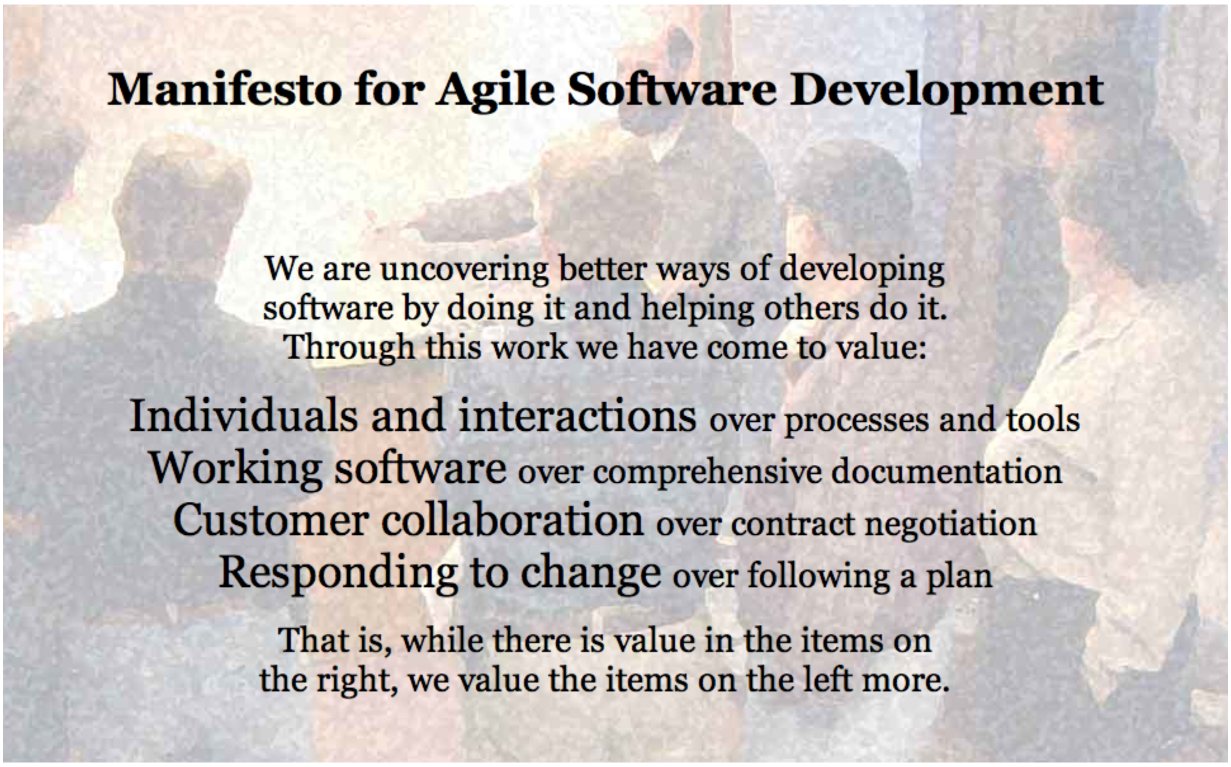
- **Business Value Game** – *Iterative planning and business value*
- **Relative Estimation** – *Basics of relative estimation*
- **Writing User Stories** – *writing effective user stories, INVEST criteria*
- **Scaling Scrum** – *Key challenges and techniques for scaling Scrum in a controlled way*
- **Technical Practices (XP)** – *What PO needs to know about technical practices*
- **Estimating Business Value** – *Relative values, point-based methods, Kano model*
- **Project Kick-Off Exercise** – *How to kick-off a project effectively, initial release plan*
- **Multi-Site Scrum** – *Key challenges and mitigation strategies for distributed Scrum*
- **Managing Maintenance** – *Control strategies for continuous flow of work*
- **Eliminating Waste** – *7 types of waste in product development, values stream mapping*
- **Agile Contracts** – *How to turn any contract into an Agile contract*
- **Pre-Assignment Wrap-Up** – *Wrapping up the pre-assignment exercises, discussion*
- **Agile Stage-Gate Model** – *Adjusting stage-gate model to accommodate Agile projects*
- **Refining User Stories** – *Research vs. development, spikes, grooming, splitting stories*
- **Traditional PM vs. Scrum Roles** – *Examining how PM role maps to Scrum roles*

Participants are also encouraged to include their own topics to the list.

There are too many potential topics that about half of them cannot be covered during the course. Additional information to the topics can be found from course material that is delivered after the course.

AGILE MANIFESTO

Here is the Agile Manifesto (for software development). In the space below the manifesto, you can create a mind map of your insights and notes about it and the discussions.



Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

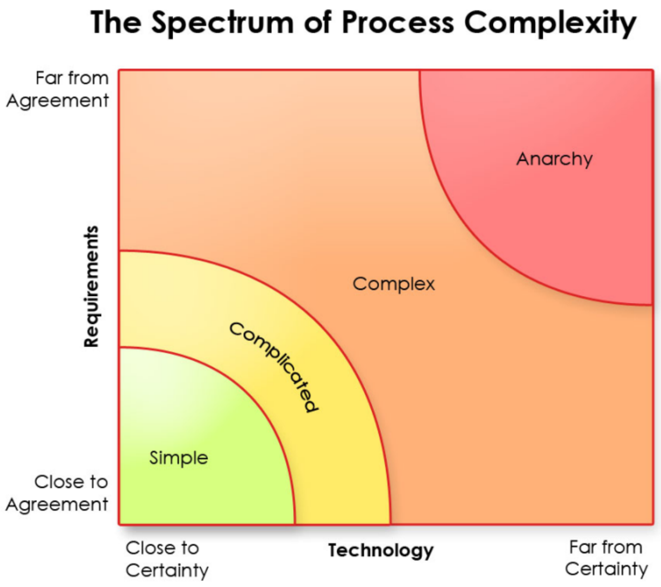
That is, while there is value in the items on
the right, we value the items on the left more.

Agile
Manifesto

COMPLEXITY AND CAUSALITY

Quoting from http://www.noop.nl/2008/08/simple-vs-complicated-vs-complex-vs-chaotic.html,

> ### My car key is simple.
> It took me about three seconds to understand how my car key works. OK, maybe that's not quite correct. Mine has a battery in it. If I take it apart it might take me another three hours to understand its details. But yeah, I'm smart, I'll manage.
>
> ### My car is complicated.
> It would take me years to understand how my car works. And I don't intend to. But if I did, then some day in the far future I would know with certainty the purpose of each mechanism and each electrical circuit. I would fully understand how to control it, and I would be able to take my car apart and reassemble it, driving it exactly as I did before. In theory, of course. In practice, only real men do things like that.
>
> ### Car traffic is complex.
> I can travel up and down the same street for twenty years, and things would be different every time. There is no way to fully understand and know what happens around me on the road when I drive, how other drivers operate their vehicles, and how the people in the streets interact. I can make guesses, and I can gain experience in predicting outcomes. But I will never know for sure.
>
> ### Car traffic in Lagos (Nigeria) is chaotic.
> When things get too complex, they easily become chaotic. Traffic in Lagos is so bad, it is not even predictable. Poor infrastructure and planning, heaps of waste, pollution, lack of security, floods, and many more problems make it one of the worst places in the world to be, as a simple car driver.

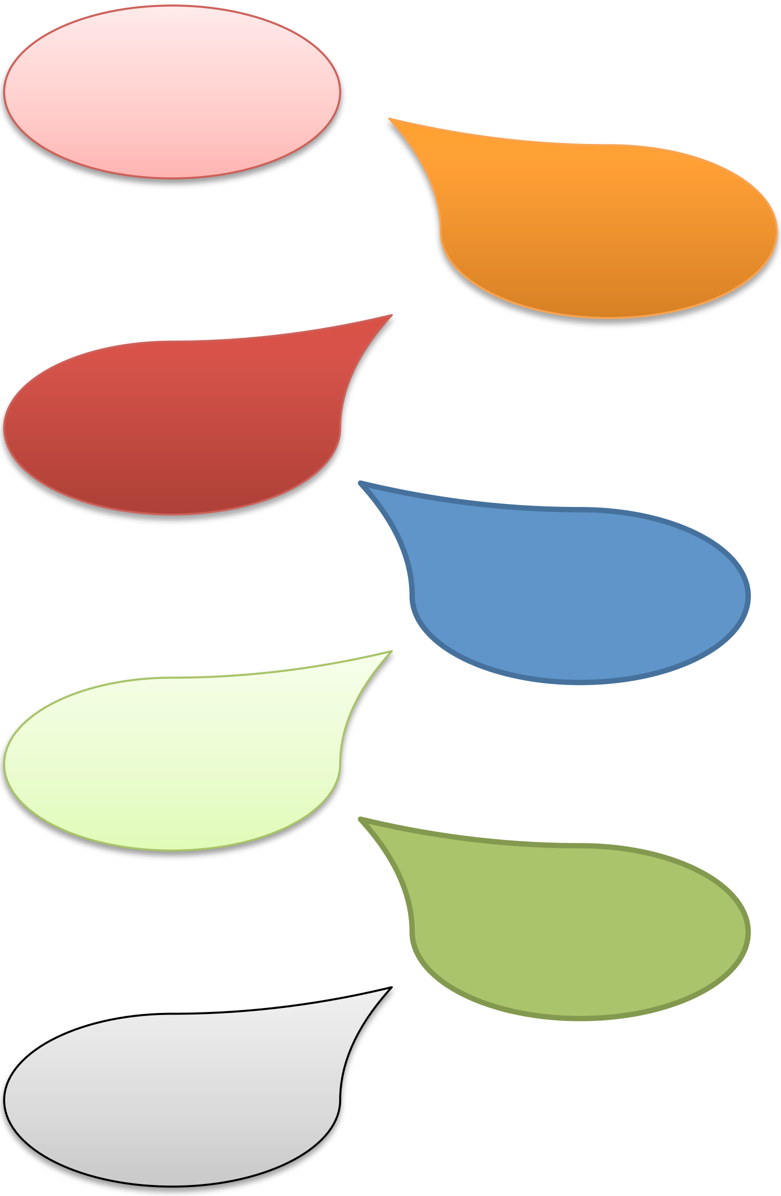


Discuss in 3-4 person groups what each of these categories mean

Then, brainstorm in 3 minutes as many examples to each of the categories as you can. Try to find examples that are related to your working environment.

ELIMINATING WASTE

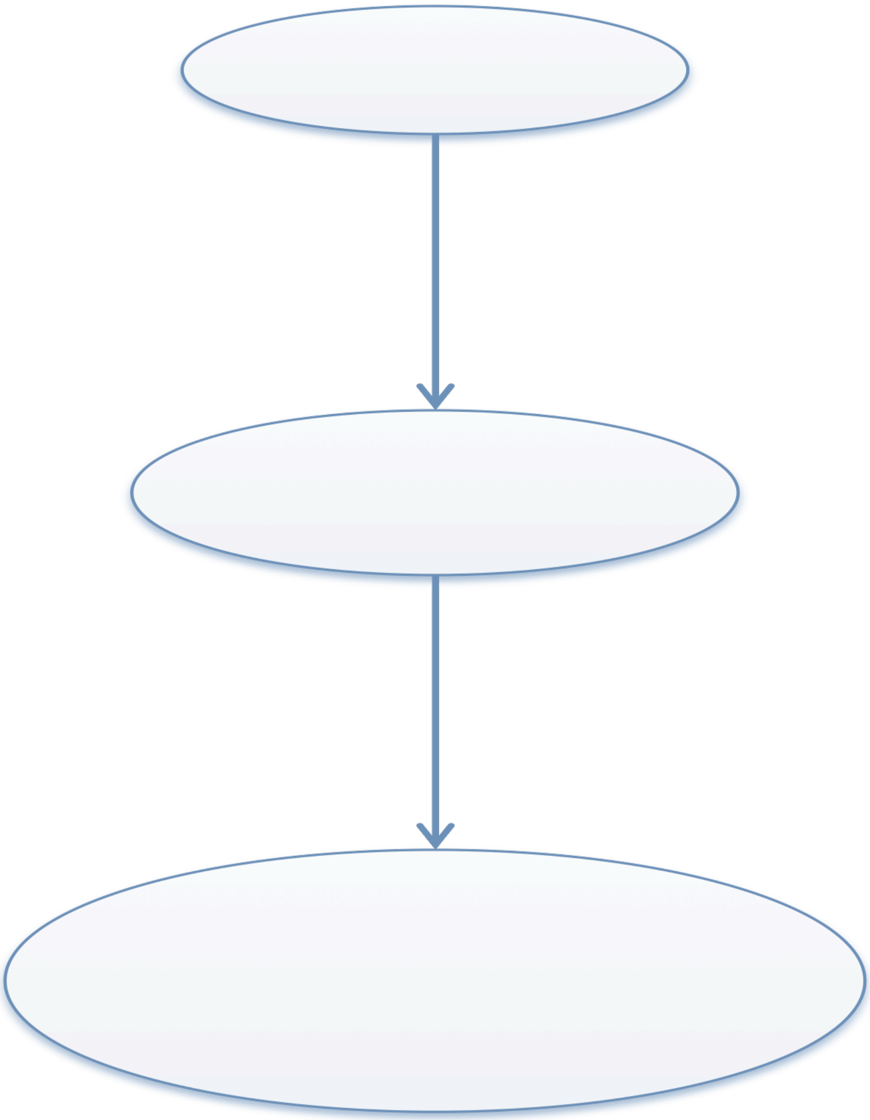
The seven categories of waste (as defined by Tom and Mary Poppendieck, in Lean Software Development):



In your table group, discuss the seven forms of waste and identify at least one example for each category. Then discuss how those wastes could be eliminated (or at least significantly reduced).

THE BIG PICTURE

As this topic is discussed, update the diagram below with conceptual levels and elements that are related to each level.



SCRUM ROLES VS. TRADITIONAL PROJECT MANAGER

In small groups, discuss the different kinds of responsibilities and activities traditional project managers typically have. Then discuss how those responsibilities and activities would map to Scrum roles. Please write your mapping to the diagram below.

Product Owner
responsibilities

ScrumMaster
responsibilities

Development Team
responsibilities

Responsibilities not
related to Scrum

PRODUCT VISION

As examples of how a vision can be effectively created and communicated, here are two different approaches/tools:

Geoffrey Moore's template from Crossing the Chasm:

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)*

Product Box

Design the cover (or the whole box) of an imaginary product box, in which the product would be shipped out in. For services or products without a box, just imagine it would be delivered in one. The box cover should have:

- The name of the product
- An image or logo of the product
- 3-4 key features (and no more than that)

User Testimonials

Write three user testimonials that you would like to hear from the actual users after the launch of the product or system. Each testimonial should have:

- One key feature or an important aspect of a central feature
- Description of the way in which the feature was useful to the user
- Name of the user and some description of the user's type or background

In your table groups, spend a few minutes discussing e.g. the following questions:

- Did the exercise work in clarifying the vision?
- What were the most important and useful part of the exercise?
- Why do exercises like these work? What is important in the exercise itself?

WRITING USER STORIES

Typical templates used:

- ***As a*** *<user role>**, I can*** *<do something>* ***in order to*** *<some benefit or purpose>*
- *<User role>* ***can*** *<do something>* ***so that*** *<some benefit or purpose>*

The “can” work can be replaced by “must”, “should”, or any appropriate verb.

Remember, Scrum talks only about “product backlog items” (PBI’s); user stories is just one way of writing a PBI. Also other types of items can be placed in the Product Backlog.

User stories consist of three parts (three C’s):

- ____
- ____
- ____

When writing user stories, the first version of the story can be virtually anything, but as the actual development iteration approaches, it is recommendable to groom them to meet the INVEST criteria:

I = ____

N = ____

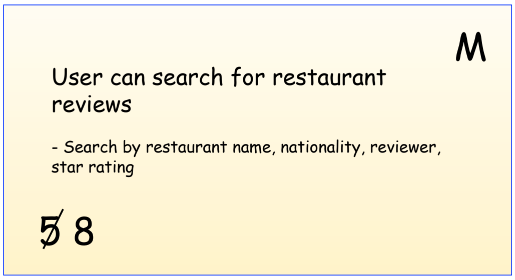
V = ____

E = ____

S = ____

T = ____

If you use index cards, it helpful to leave some space to the top and bottom of the card for different attributes, e.g. MoSCoW priority, value, size estimate.



M

User can search for restaurant reviews

- Search by restaurant name, nationality, reviewer, star rating

~~5~~ 8

REFINING USER STORIES

Refining user stories involves adding detail and acceptance criteria to them, and breaking them smaller. We'll focus on the last one here.

When splitting stories, it is very important to maintain the user-focused approach. It is not allowable to break stories into the activity or modular components (tasks).

There are generally two types of stories:

- Compound stories consist of multiple easily splittable substories
- Complex stories are typically large complicated features that are often difficult to break down

Stories can typically be split along:

- ______________________________, e.g. _______________________________________
  ____
- ______________________________, e.g. _______________________________________
  ____
- ______________________________, e.g. _______________________________________
  ____

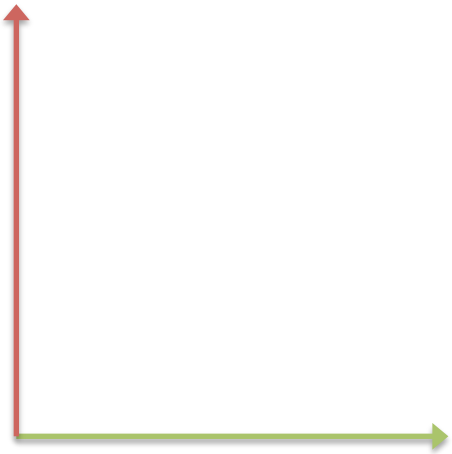
Write below any other notes you want to make while discussing the topic and doing the story refinement exercise.

PROJECT KICK-OFF

To get a project started, it needs a product backlog that has been sufficiently prioritized to allow the selection of valuable work for the first sprint. One tested sequence of analysis and planning activities, preceded by user story writing, is the following (write your comments and notes to the space below the names of the practices):

MoSCoW

Risk / Value Estimation



Dependency Mapping

Initial Release Plan

RELEASE PLANNING

Let us assume the following simplified Product Backlog (105 story points in total size):

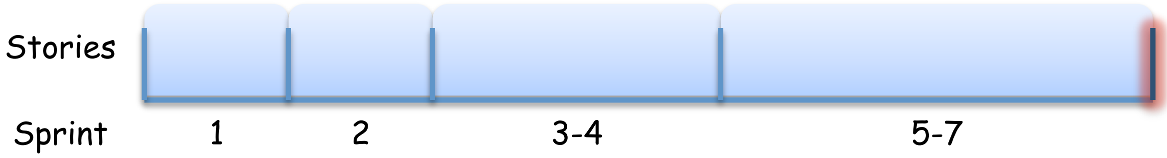
Story number	Size
Story 1	3
Story 2	2
Story 3	5
Story 4	5
Story 5	3
Story 6	8
Story 7	5
Story 8	13
Story 9	13
Story 10	20
Story 11	8
Story 12	20

The team has been working on the product for already several sprints and has established probable velocity between 7 and 10. The next release is 7 sprints into the future.

- What is the scope that the PO can commit to stakeholders with reasonable safety?
- Which stories can pretty much be ruled out of probable release scope?

Draw two lines into the Product Backlog to indicate the above two scope limits.

Map the stories (using story numbers) to the following simple release plan.



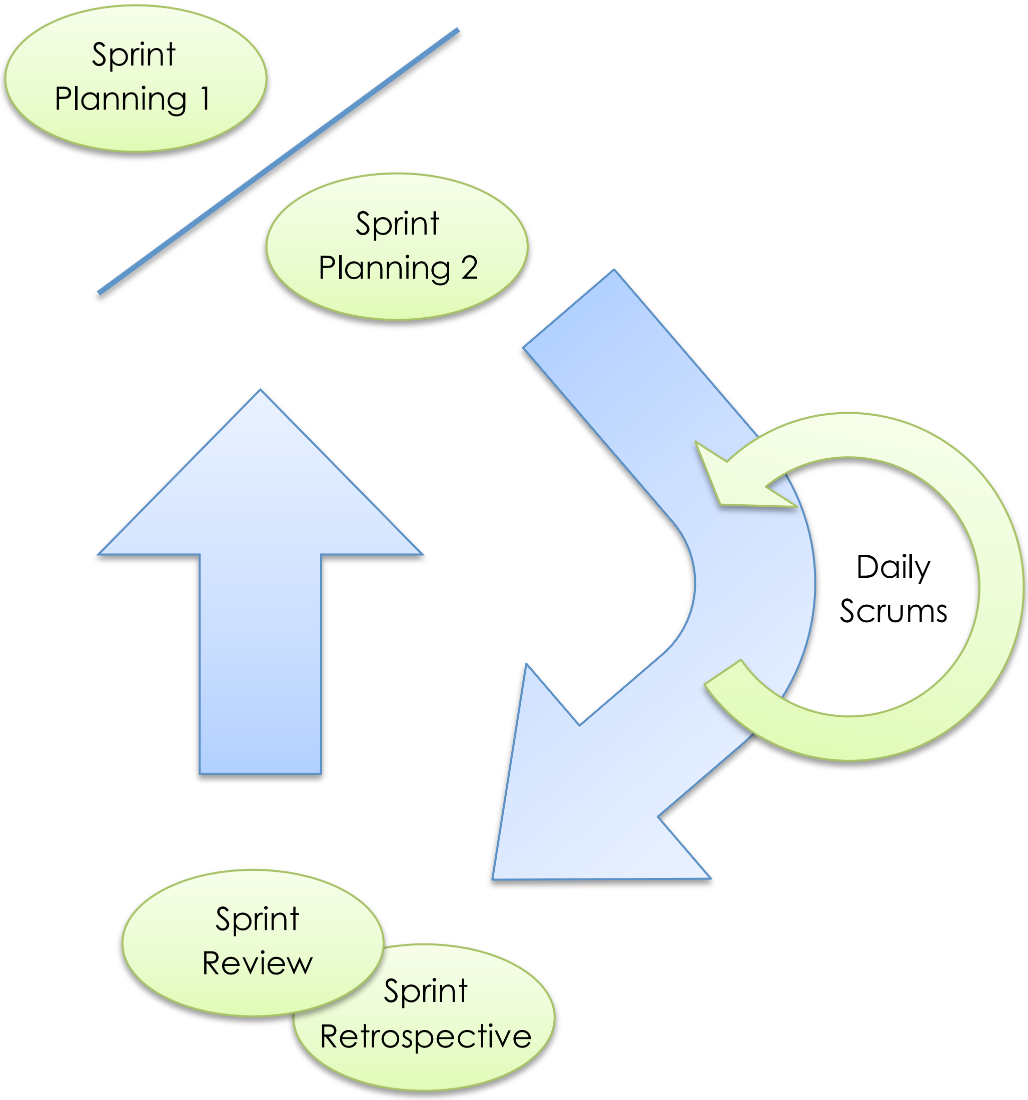
- How many points worth of stories can you allocate to each sprint "slot"?
- Is this simple method sufficient for release planning? If not, what other things need to be considered?

Consider the following scenarios:

- 2 Sprints forward, the team has delivered up to story 6 (26 more story points). They've updated their probable velocity to between 9 and 14. How does this change the release plan?
- User review at the end of Sprint 2 revealed new ideas and requirements that should be incorporated into the release. The team has estimated that their total size is 16. How does this change affect the release plan?

SPRINT PLANNING AND EXECUTION

The iterative and incremental product development cycle in Scrum is managed in four key meetings – Sprint Planning, Daily Scrums, Sprint Reviews, and Sprint Retrospectives. As we discuss them in more detail, please do write your key insights into the diagram below.



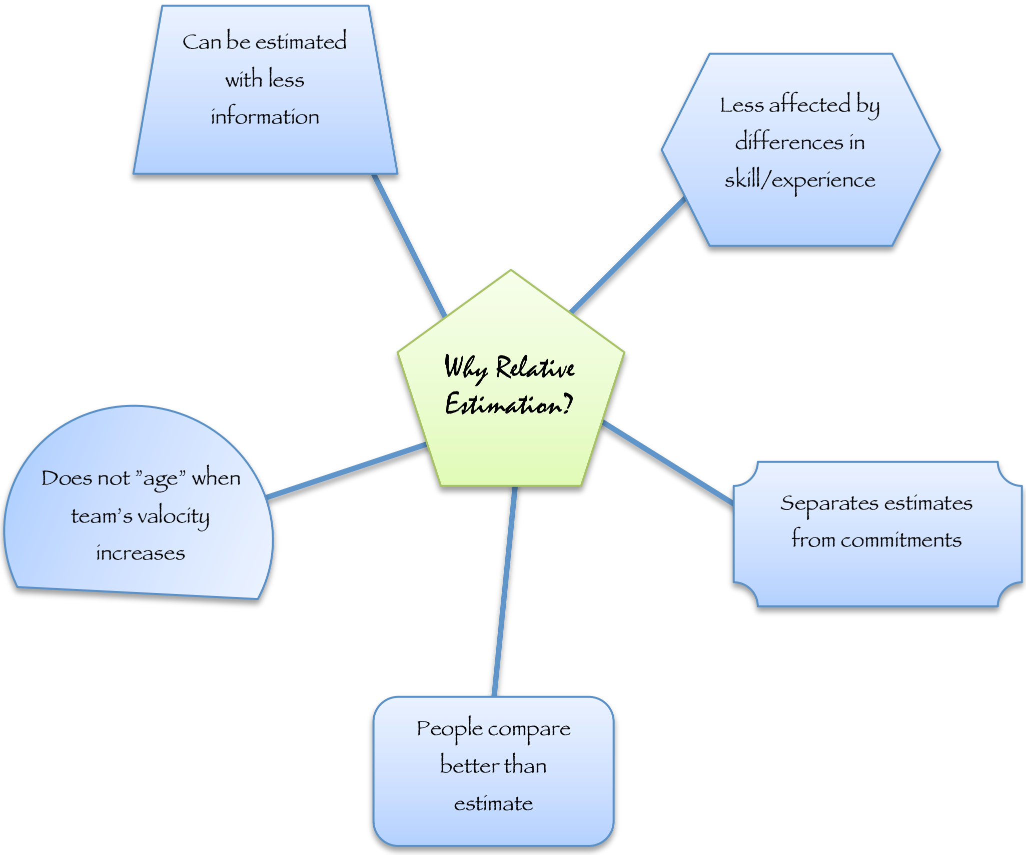
Compare your notes with a pair. What kind of differences do you notice? What do you make of those differences?

RELATIVE ESTIMATION

In relative estimation, items are compared against one another rather than against a certain metric. For example, one item maybe considered twice the size of another, rather than estimating how many hours or days it would take to do.

In the diagram below, add any additional notes you think are relevant to this topic.

In table groups, discuss each of the five benefits. What do each of them mean? Do you know some data or information regarding them?

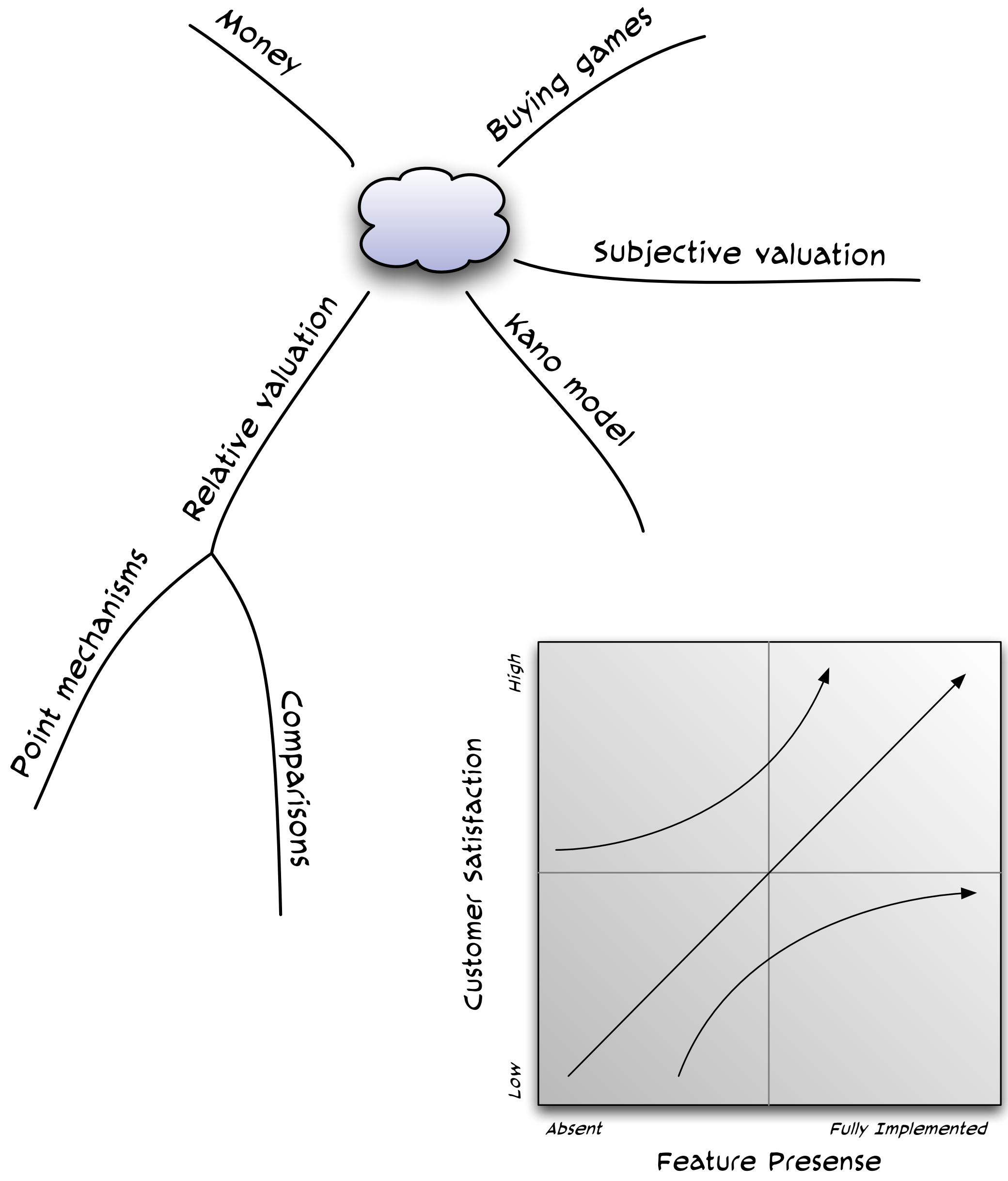

Why Relative Estimation?
Can be estimated with less information
Less affected by differences in skill/experience
Does not "age" when team's valocity increases
Separates estimates from commitments
People compare better than estimate

When would time-based estimates be more useful? Where using them would make more sense than relative estimation?

ESTIMATING BUSINESS VALUE

If you can estimate the monetary value of a feature or a product backlog item, it is often recommendable to use those values for value estimates. However, most of the times it’s difficult to pinpoint sufficiently meaningful monetary values and other techniques need to be used.

Write below your comments while we discuss various alternative techniques.



The following simple comparison technique can be used to weigh different themes / features against one another using various selection criteria. One theme / feature is selected as a baseline (gets 0 for all criteria), and other themes are compared against it. If the compared theme / feature is better than the baseline, it gets "+". If it's worse, it gets "-". If they are roughly equal, it gets "0". The relative value of the theme / feature is the sum of plusses and minuses.

Theme Screening Worksheet

MOUNTAIN GOAT SOFTWARE

		Themes						
Selection Criteria								
	Net score							
	Rank							
	Continue?							

+ = Better than 0 = Same as – = Worse than

Copyright - Mountain Goat Software, used with permission

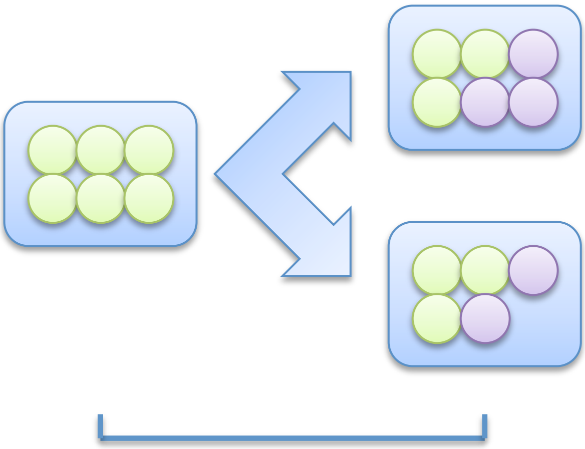
MANAGING MAINTENANCE

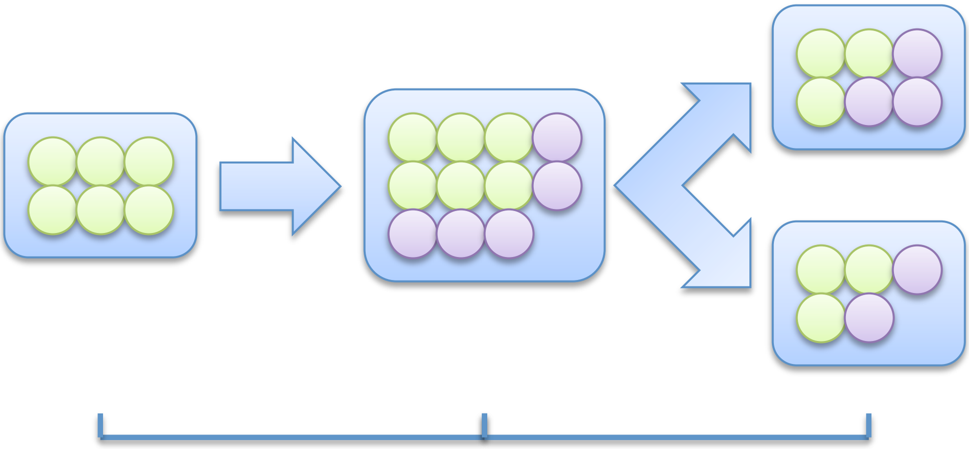
Scrum is fundamentally a batch-driven approach. It works best when work can be allotted into those batches. However, many environments have rapidly changing priorities and fast response times, and it's impossible to wait for the next Sprint to act on that in a managed way. Fortunately, there are excellent continuous flow frameworks available.

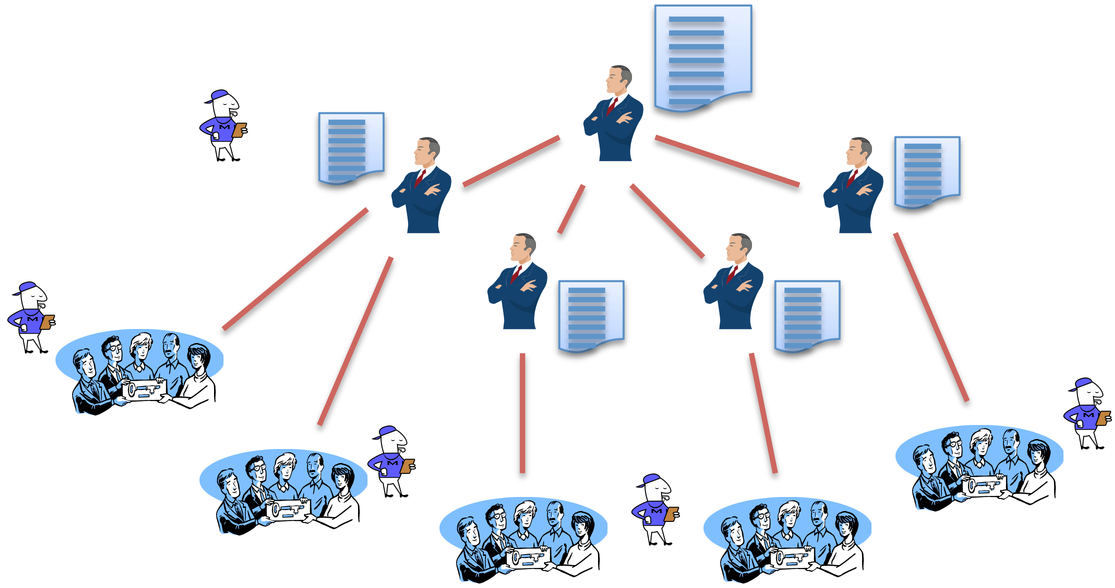
In the space below, please make notes as we discuss the issue and look at the Kanban framework. You can also draw yourself a simple Kanban board.

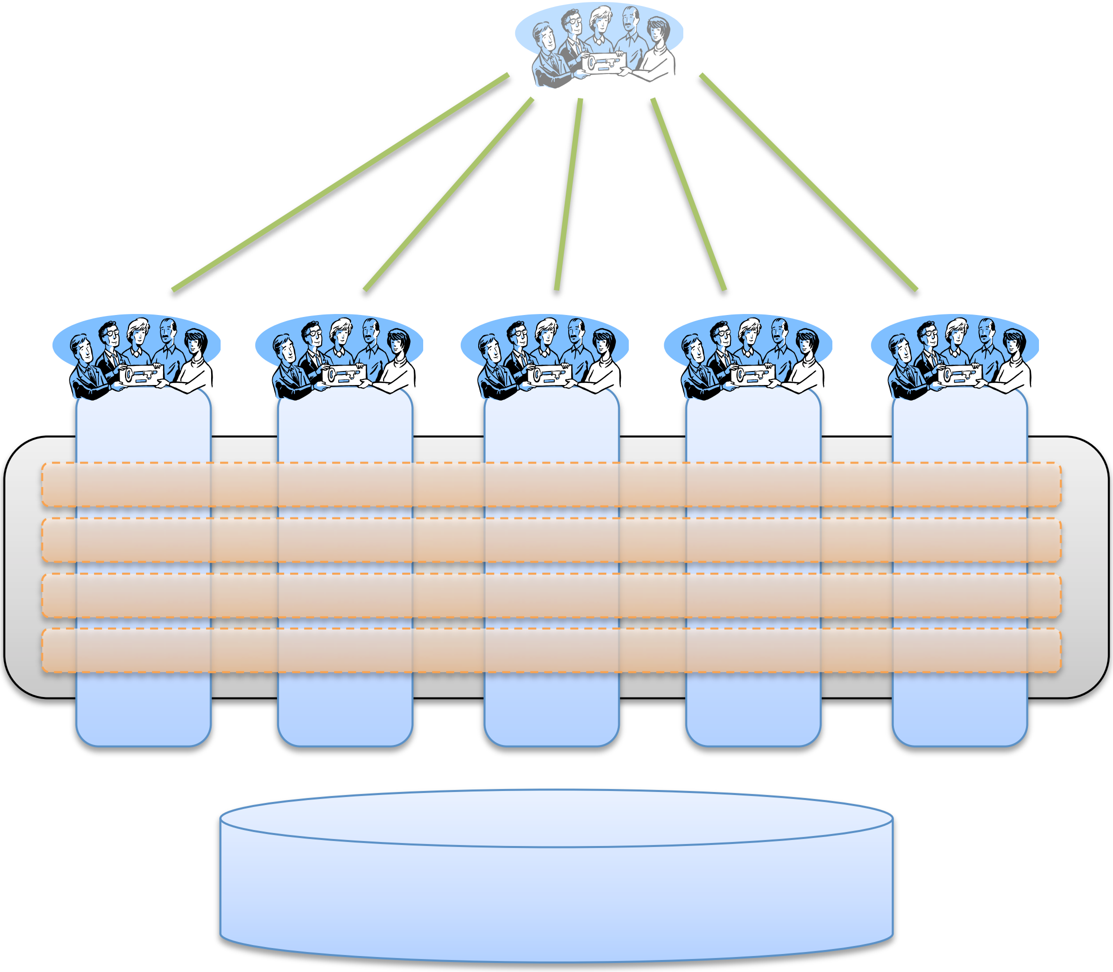
SCALING SCRUM

Please add your comments regarding scaling to the diagrams below.









MULTI-SITE SCRUM

In your table group, discuss problems caused by distributing Scrum teams and projects across multiple sites and possibly across multiple time zones.

There are many ways these problems can be mitigated, but the reduced productivity from distribution is very difficult to remove entirely. As we discuss the various possible techniques, write down the key mechanisms below.

Mitigating Multisite

AGILE CONTRACTS

In my experience, Agile software development can be done using largely the same contracts as traditional development. However, there are certain things that is recommended to be different. The following additions can be added to virtually any contract.

DEFINITIONS

We mutually agree on working together, and thereby build trust in each others expertise.

Customer Participation in Scrum Team:

The Customer is expected to be active in the project. The role of the Customer includes the following:

- Prioritize features by business value and have them implemented in order of maximum value
- Mutually agreed estimates for all work items. The official representatives of both parties need to agree. This needs to be noted in a signed addendum to the contract for each change.
- Participate in each Sprint planning meeting by discussing the selected features with Company team, including answering questions to provide clarification to the team.
- Participate in writing the conditions of satisfaction for each feature, so the team and client have a shared definition of when a feature is done. These conditions should be completed as part of a user story before any code is written.
- Participate in each Sprint review meeting, and provide timely feedback both for both work-in-progress and completed work.

***To company: Things to ensure are defined in the contract:

- Total value of the contract
- Rates for time and materials billing
- Scope of the contract

CLAUSE: EARLY TERMINATION (MONEY FOR NOTHING)

The Customer may terminate the contract at the end of any Sprint. The standard metric for termination is when the Customer perceives the cost of continuing the project is higher than the additional value received. The Customer will pay Company 20% of the remaining contract value to exercise early termination.

Company commits to delivering 80% of the project scope as high quality by the agreed upon delivery date. High quality is defined by the agreed upon Definition of Done.

This clause can only be enacted if the Customer maintains Participation in the Team Scrum during the project.

In the event that both parties cannot mutually agree on work item estimates or that the Customer does not maintain participation in the Scrum Team, the contract shall revert to a time and materials billing.

CLAUSE: CHANGE FOR FREE

If the Customer maintains Participation in Scrum Team during the entire project, Customer shall be able to make changes to the Scope without incurring any additional cost if total Scope of contracted work is not changed. New features may be added for free at Sprint boundaries if items of equal scope are removed from the contract.

http://www.coactivate.org/projects/agile-contracts/money-for-nothing-change-for-free

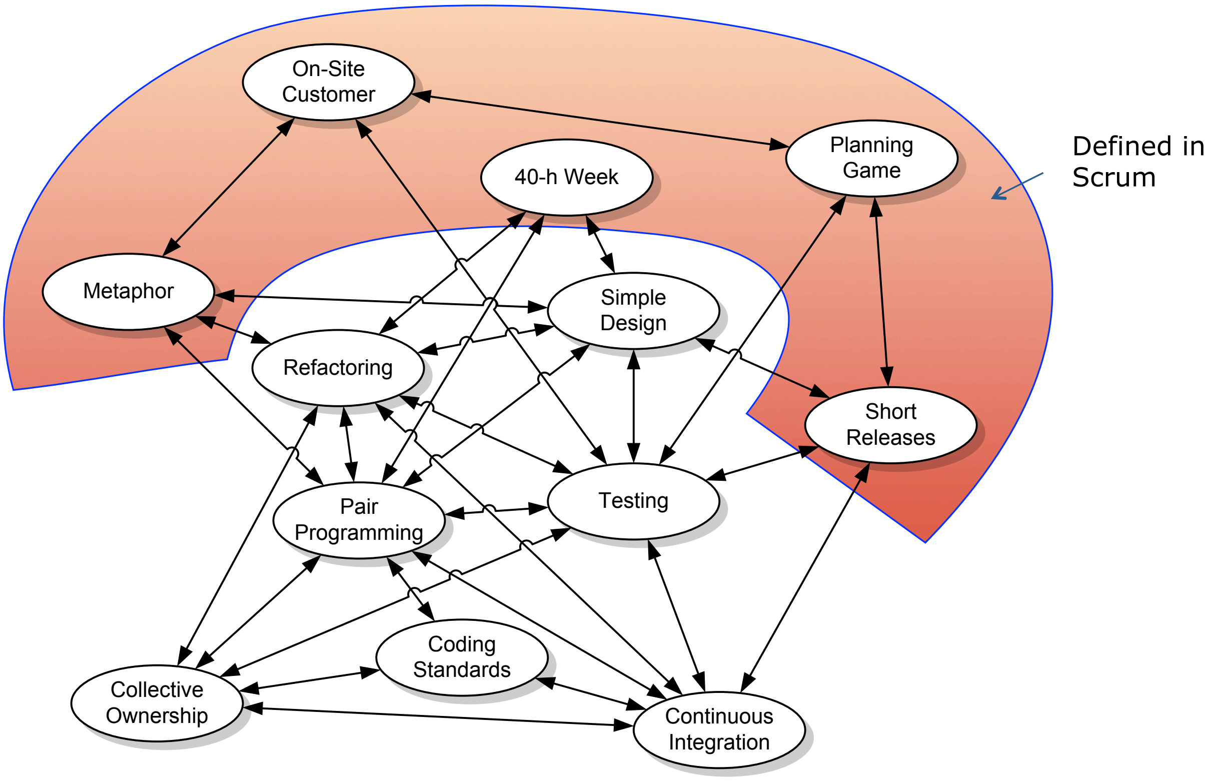
In your table group, discuss the impact these two clauses would have on a project contract.

TECHNICAL PRACTICES (XP)

Extreme Programming (XP) forms a solid methodology for development of software systems. Unlike Scrum, it is therefore tied to software development domain. Given that it is assumed that a Scrum team will seek ways in which they can effectively deliver a new tested version of the system in every sprint, it is expected that the team adopt XP or similar technical practices through continuous inspect and adapt cycles. Unfortunately, it is not always the case.

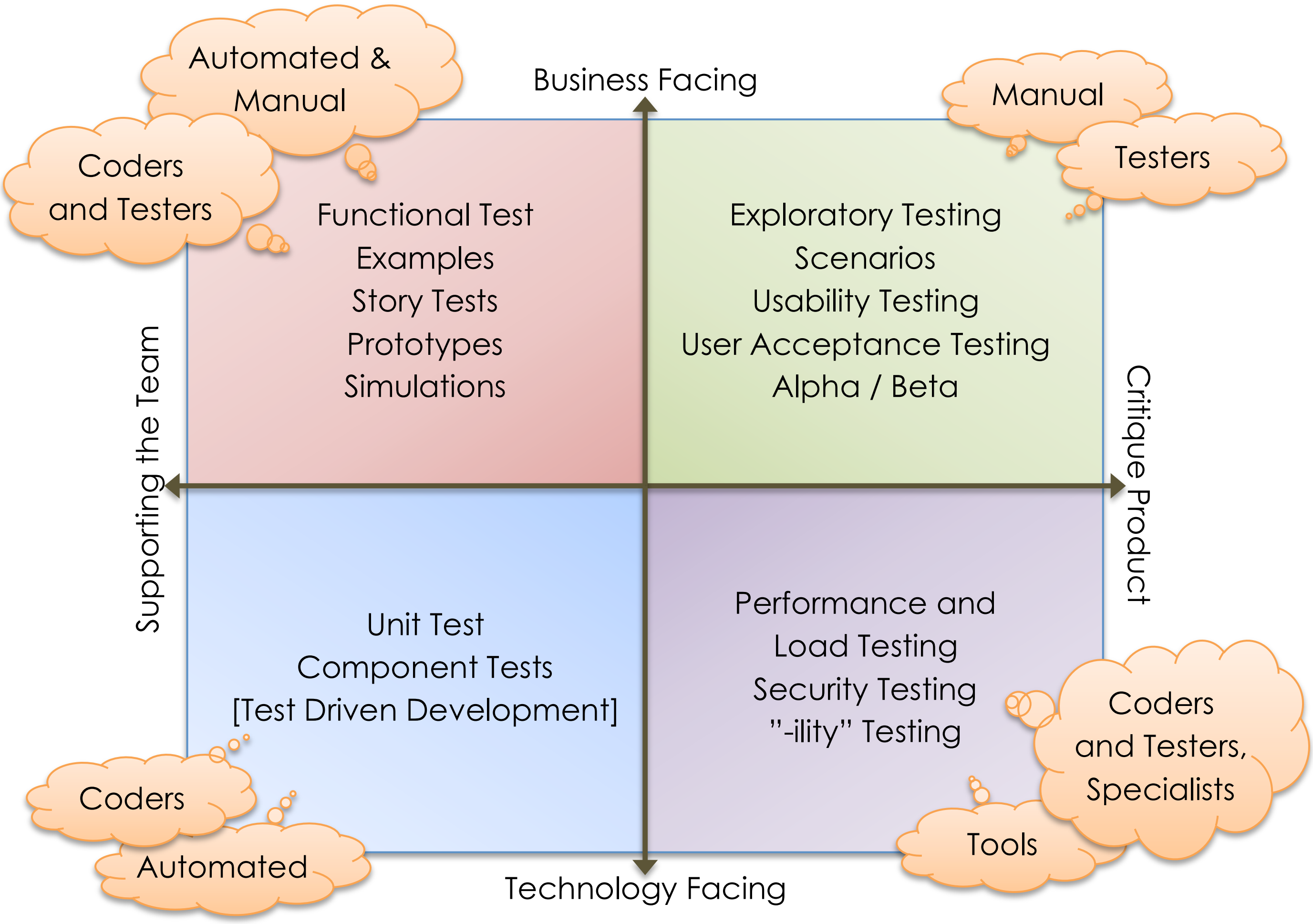
XP consists of a set of development practices, each of which is simple and insufficient in itself, but becomes very powerful when combined with and supported by other XP practices.

The practices, and their key dependencies, are:



> "I've never seen or heard of a hyperproductive team that wasn't doing the eXtreme Programming practices (as described by Kent Beck, Ron Jeffries, etc.)."

Michael James, http://danube.com/system/files/A_ScrumMaster's_Checklist_blog.pdf

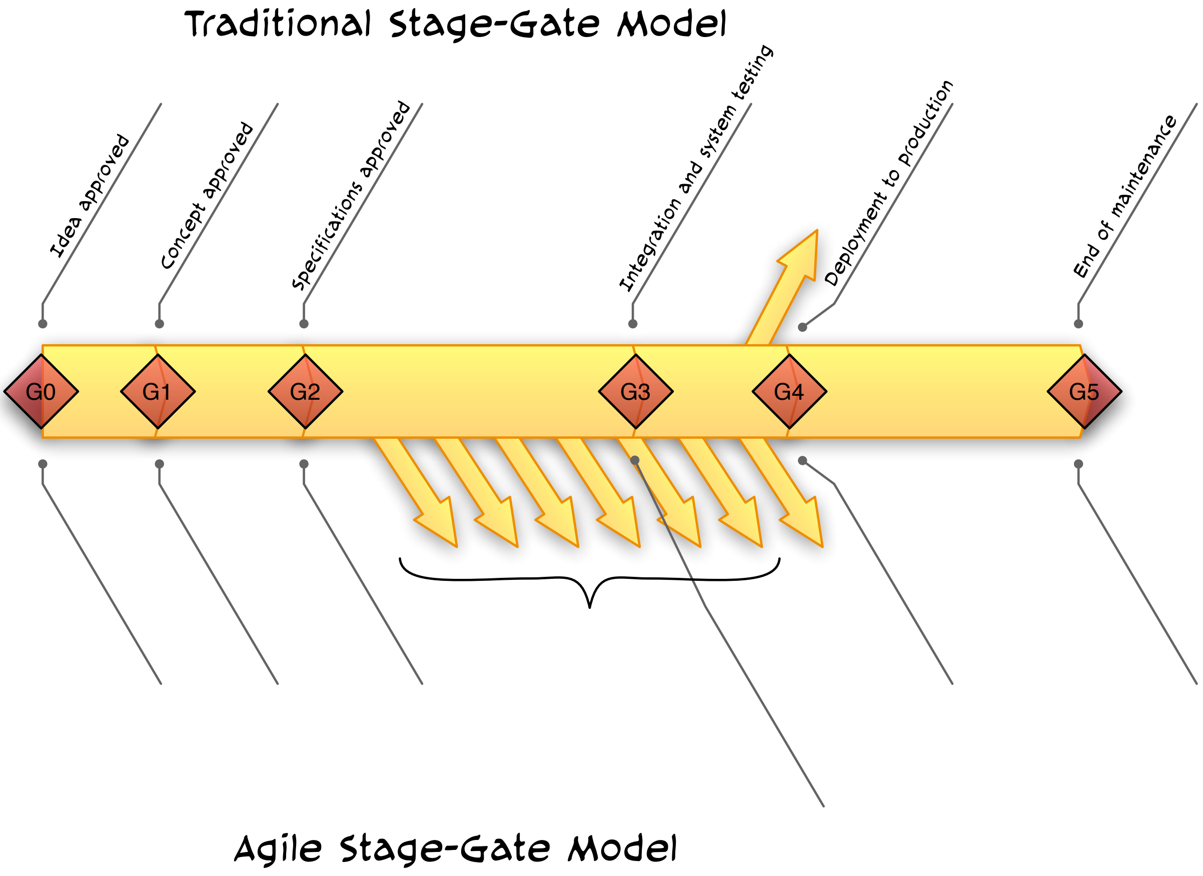


Adapted from Janet Gregory's version of Brian Marick's original diagram

AGILE STAGE-GATE MODEL

Many organizations, while transitioning from traditional project management to Agility, need to reconcile the problem of managing both approaches in project portfolio management level. The traditional governance approach has been to use stage-gate models with clearly defined inspection and approval points. Agile projects can comply with the intent of the model, but some of the points need to be treated differently.

As we discuss, write down your comments in the diagram below.



PRE-ASSIGNMENT WRAP-UP

Use the space below for any comments that you want to write down during pre-assignment wrap-up.

PETRI HEIRAMO

CollabNet Certified Scrum Trainer (CST) and Agile Transformation Mentor

I'm particularly interested in looking at Scrum and Agile as a holistic approach to improving the competitiveness of businesses. I find myself often in discussions where I emphasize the need for businesses to focus on generating value (over managing costs) as a means to success. Be it in a project or in an organization, I frequently see dysfunctions where silly savings are made at the cost of significant value. And many times the decision makers are totally unaware of the impact of their decisions. It is this mismatch between intent (to work for the best of the company) and action that I seek to eliminate. When focusing on value, we need to focus on customer delight, job satisfaction and continuous improvement.

I've studied Agile methods actively since Fall 2005. In addition, to leading Scrum and Agile projects, I have been doing Scrum and Agility training and consulting since early 2006. I was certified as Scrum Practitioner in April 2007 and as CST in November 2008.

I have over 10 years of experience in developing software as a developer, project manager, QA manager, ScrumMaster, process developer, coach and trainer – and experience with both the traditional and Agile approaches to software development. I've seen a wide variety of projects, and as a result, have come to appreciate the positive effects of Scrum on the human side of software development. Seeing people enjoy their work again has been my greatest joy in Agile deployments. Of course, seeing a happy and engaged customer is great, too, and so is instilling a new kind of pride and common sense in the work people do.

Transform with Experience

CollabNet's ScrumCORE™ training division has helped hundreds of companies successfully adopt and scale Scrum and Agile. As the largest Scrum Alliance certified training facility, CollabNet's Certified Scrum Trainers possess deep experience leading organizations — from small businesses to multinational enterprises — through Agile transformations. With a full range of training services, CollabNet's ScrumCORE offerings can meet the needs of any organization.

The ScrumCORE team conducts regularly scheduled public courses in major markets throughout North America and Europe, offering ScrumMaster and Product Owner Certification courses. Additionally, CollabNet's ScrumCORE trainers are available for on-site private coaching engagements. These coaching sessions connect organizations with an experienced Scrum coach, who can lead them through a transformation or help them resolve specific challenges.

For more information, visit http://www.open.collab.net/training/scrummaster, or to speak with a training specialist contact scrumtraining@collab.net.

CONTACT US

Corporate Headquarters
8000 Marina Blvd,
Suite 600
Brisbane, CA 94005
United States
Phone: +1 (650) 228-2500
Toll Free: +1 (888) 778-9793

Chennai, India Office
Phone: +91 44 4220-3700

Shanghai, China Office
Phone: +86-21-61221082

Seoul, Korea Office
Phone:+82-2-722-8271

Tokyo, Japan Office
Phone:+81-3-5789-5366

London, UK Office
Phone:+44 (0) 207-397-8690

Munich,Germany Office
Phone: +49 (89) 24218-442