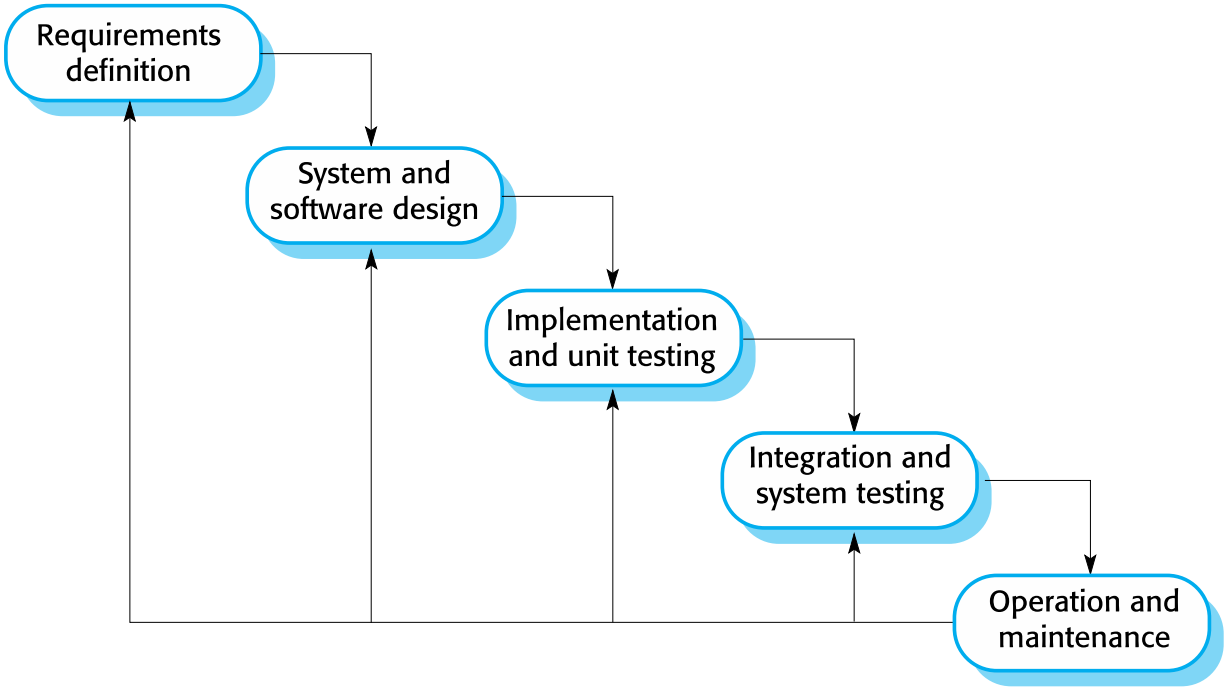
Introduction to Scrum

Chien-Hung Liu
Dept. of CSIE, Taipei Tech
6/15/2023

Part of the slides adapted from Marty Stepp, Scrum Primer, Mountain Goat LLC, Scrum Alliance and The Scrum Guide™

Traditional Software Development

- The traditional way to build software was a sequential life cycle commonly known as waterfall
  - Plan and schedule all of the process activities before starting software development



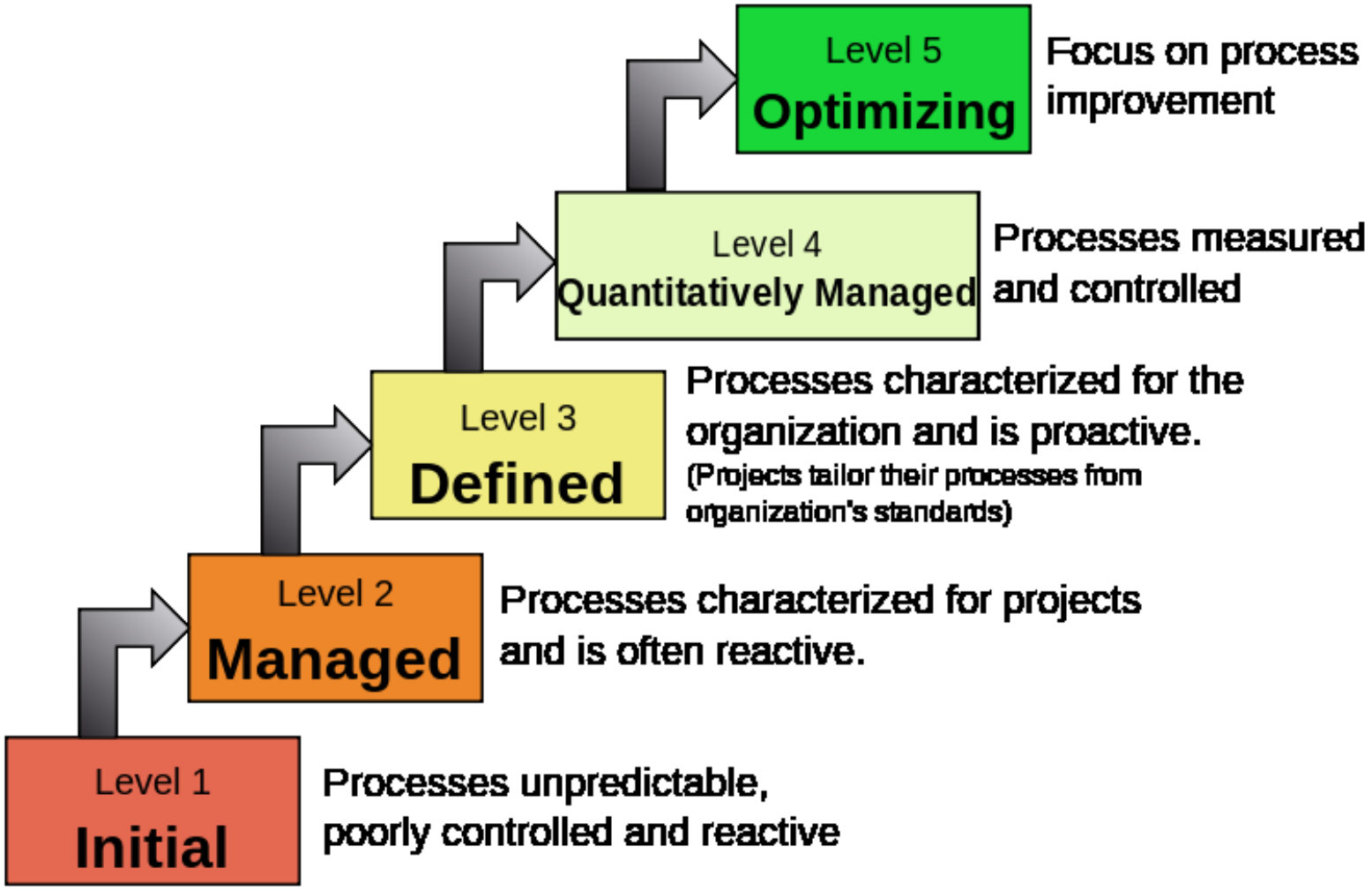
The Waterfall Process

- It is supremely logical, think before you build, write it all down, follow a plan, and keep everything as organized as possible
  - Good for large and long-life systems in large organizations

- **Plan-driven** process requires that the good ideas all come at the beginning of the release cycle, where they can be incorporated into the plan
  - This also makes it difficult to accommodate changes after the process is underway
- It also places a great emphasis on writing things down as a primary method for communicating critical information
  - Documents have merits. However, engineers are not good at writing documents. Incomplete, ambiguous, and inconsistent documents can also cause problems and waste efforts

CMMI (Capability Maturity Model Integration)

Characteristics of the Maturity levels



The Agile Manifesto

Individuals and interactions	over	Process and tools
Working software	over	Comprehensive documentation
Customer collaboration	over	Contract negotiation
Responding to change	over	Following a plan

12 Agile Principles 1/2

1. Our highest priority is to satisfy the customer through **early and continuous delivery of valuable software**
2. **Welcome changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage
3. **Deliver working software frequently**, from a couple of weeks to a couple of months, with a preference to the shorter timescale
4. **Business people and developers must work together** daily throughout the project
5. Build projects around **motivated individuals**. Give them the environment and support they need, and trust them to get the job done
6. The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**

12 Agile Principles 2/2

7. **Working software** is the primary measure of progress
8. Agile processes promote **sustainable development**. The sponsors, developers, and users should be able to maintain a constant pace indefinitely
9. Continuous attention to **technical excellence** and good design enhances agility
10. **Simplicity**–the art of maximizing the amount of work not done–is essential
11. The best architectures, requirements, and designs emerge from **self-organizing teams**
12. At regular intervals, the team reflects on how to become more effective, then **tunes and adjusts** its behavior accordingly

Scrum

- Scrum原始意義是橄欖球運動中的列陣爭球
- Sprint的意義是衝刺
- Developed in early 1990 by
  - Ken Schwaber
  - Jeff Sutherland

- Scrum
  - taking a short step of development
  - **inspecting** both the resulting product and the efficacy of current practices
  - **adapting** the product goals and process practices
  - *Repeat forever*

Definition of Scrum

- Scrum is a **framework** for developing, delivering, and sustaining complex products
  - A **framework** within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value

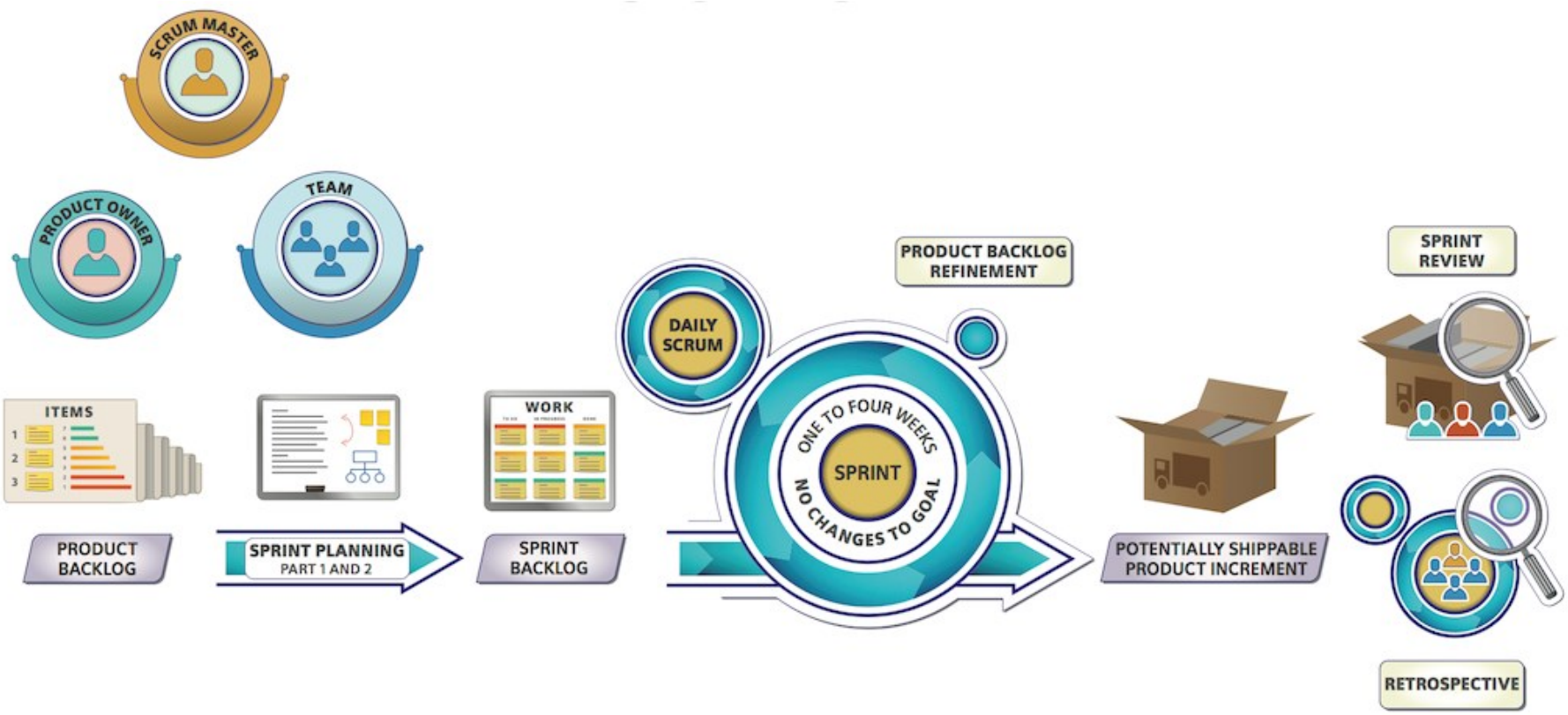
- Scrum is:
  - Lightweight
  - Simple to understand
  - **Difficult to master**

- Scrum is not a process, technique, or definitive method. Rather, it is a framework within which you can employ various processes and techniques

What is Scrum

- Scrum is a development framework in which **cross-functional teams** develop products or projects in an iterative, incremental manner
  - Structures development in cycles of work called **Sprints**
    - Each sprint is **time-boxed** and **never extended** (no more than 4 weeks)
  - A cross-functional **Team** selects **items** (customer requirements) from a prioritized list
    - During the Sprint, **no** new items may be added
    - Scrum embraces change for the *next* Sprint
  - Every day, the Team gathers briefly to **inspect** its progress, and **adjust** the next steps needed to complete the work remaining
  - At the end of the Sprint, the Team **reviews** the Sprint with stakeholders, and demonstrates what it has built

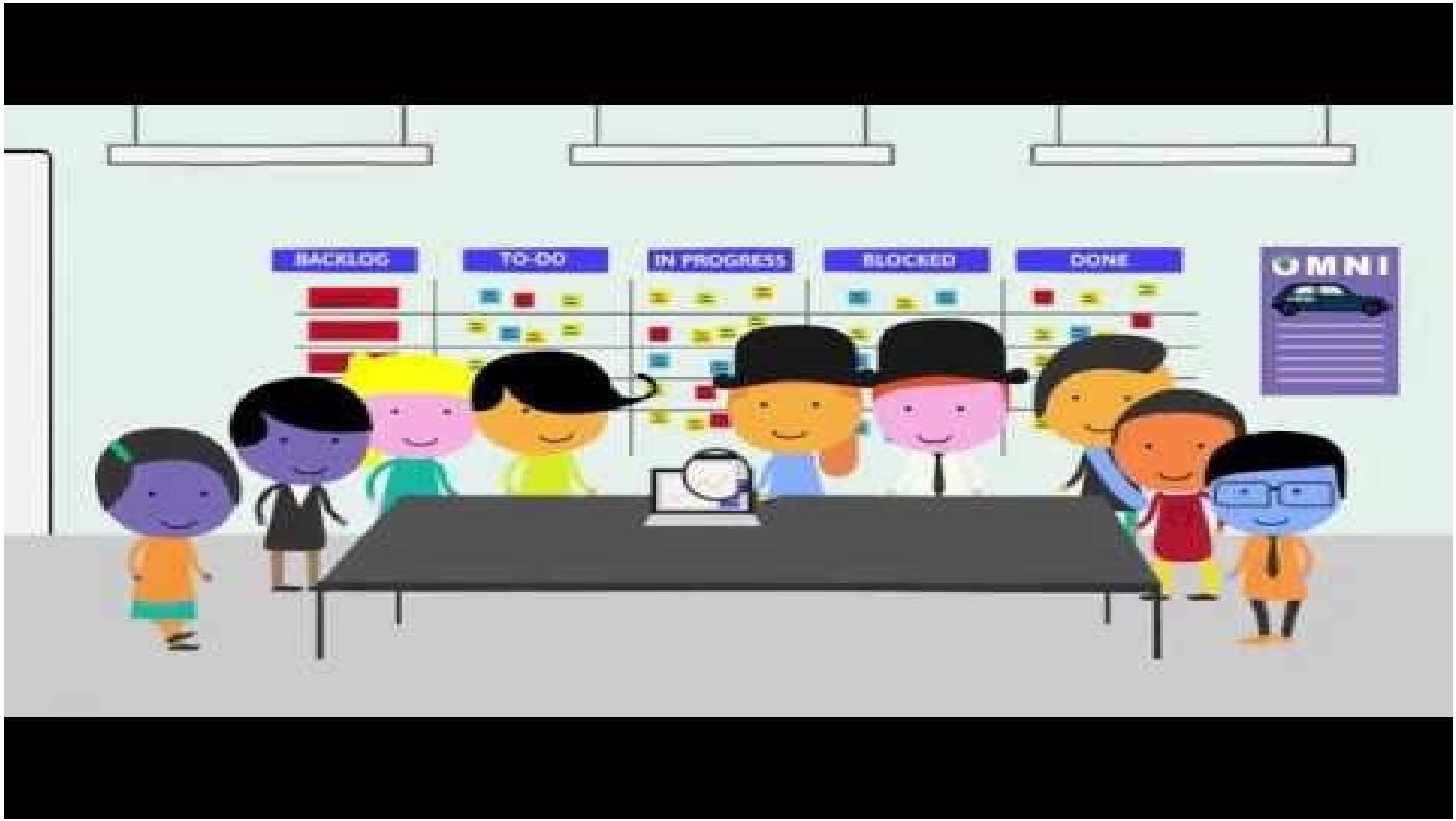
Scrum at a Glance



Source: Adapted from *Scrum primer* https://scrumprimer.org/.

A Brief Overview of the Scrum Framework

https://youtu.be/gy1c4_YixCo (2:21)



Scrum Framework

Roles

- Product owner
- Scrum Master
- Team

Events

- Sprint planning
- Daily scrum meeting
- Backlog refinement
- Sprint review
- Sprint retrospective

Artifacts

- Product backlog
- Sprint backlog
- Product Increment

Scrum Roles

- **Scrum Team** = Product Owner + ScrumMaster + Team
  - Product Owner
    - Decides features, release date, prioritization, values($$$)
    - Possibly a Product Manager or Project Sponsor
    - Accept or reject work results

  - ScrumMaster
    - Is a servant-leader for the Scrum Team
    - Responsible for enacting Scrum values and practices
    - Remove impediments / politics, keeps everyone productive

  - Development Team
    - 5-9 members ()
    - Teams are self-organizing and cross-functional
    - Membership should change only between sprints

Product Owner

- Product Owner (PO)
  - is responsible for maximizing the **value of the product** resulting from work of the Development Team (or maximizing **return on investment**, ROI by identifying product features)
  - is the sole person responsible for managing the Product Backlog
    - Clearly expressing Product Backlog items
    - Ordering the items in the Product Backlog to best achieve goals and missions
    - Optimizing the value of the work the Development Team performs
    - Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Scrum Team will work on next
    - Ensuring the Team understands items in the Product Backlog to the level needed
  - The Product Owner may do the above work, or have the Development Team do it. However, the Product Owner remains accountable

ScrumMaster

- Scrum Master
  - is responsible for promoting and supporting Scrum
    - by helping everyone understand Scrum theory, practices, rules, and values
    - help lead the organization through the often difficult change required to achieve success with agile development
  - is a coach and teacher
    - educates, coaches and guides the Product Owner, Team and the rest of the organization in the skillful use of Scrum
  - is not the manager of the Team members, nor are they a project manager, team lead, or team representative
  - does not tell people what to do or assign tasks – they facilitate the process, supporting the Team as it organizes and manages itself
  - cannot be the same individual of Product Owner
    - micro-managing can hinder team self-managing
  - should be a dedicated full-time ScrumMaster, although a smaller Team might have a team member play this role

The Team

- The Team (also called Development Team)
  - 7±2 members (co-locate, no job titles)
  - Self-organizing (self-managing)
    - choose how best to accomplish their work (to achieve Sprint goal)
    - decide how many items (from the set offered by the Product Owner) to build in a Sprint
  - Cross-functional
    - have all competencies (analysis, development, testing, UI design, …) needed to accomplish the work
  - Multi-learning
    - each team member certainly has special strengths, but also continues to learn other specialties
    - each team member will have primary, secondary and even tertiary skills (go to where the work is)
  - Team avoids multi-tasking across multiple products or projects, to avoid the costly drain of divided attentions and context-switching
  - Accountability belongs to the Team as a whole

Where is the Project Manager in Scrum?

- There is **no** role of project manager in Scrum at all (because none is needed)
- The traditional responsibilities of a project manager have been divided up and reassigned among the three Scrum roles, and mostly to the Team and Product Owner, rather than to the ScrumMaster
- Sometimes an (ex-)project manager can step into the role of ScrumMaster, but
  - the success of this approach is heavily dependent on the individual, and
  - how well they understand the fundamental difference between the two roles, both in the day-to-day responsibilities and in the mindset required to be successful

Other Stakeholders

- In addition to these three roles, there are other stakeholders who contribute to the success of the product, including managers, customers and end-users

- Some stakeholders such as functional managers may find their role, while still valuable, changes when adopting Scrum. For example:
  - They support the Team by respecting the rules and spirit of Scrum
  - They help remove impediments that the Team and Product Owner identify
  - They make their expertise and experience available
  - These individuals replace the time they previously spent playing the role of "nanny" (assigning tasks, getting status reports, and other forms of micromanagement) with time as "guru" and "servant" of the Team (mentoring, coaching, helping remove obstacles, helping problem solve, providing creative input, and guiding the skills development of Team members)

Scrum Framework

Roles

- Product owner
- Scrum Master
- Team

Events

- Sprint planning
- Daily scrum meeting
- Backlog refinement
- Sprint review
- Sprint retrospective

Artifacts

- Product backlog
- Sprint backlog
- Product Increment

The Sprint

- The heart of Scrum is a Sprint, a **time-box** of one month or less during which a "Done", useable, and potentially releasable product Increment is created
  - A new Sprint starts immediately after the conclusion of the previous Sprint
  - Sprints contain and consist of the Sprint Planning, Daily Scrums, the development work, the Sprint Review, and the Sprint Retrospective
- During the Sprint:
  - No changes are made that would endanger the Sprint Goal
  - Quality goals do not decrease
  - Scope may be clarified and re-negotiated between the Product Owner and Development Team as more is learned
- Cancelling a sprint
  - A Sprint can be cancelled before the Sprint time-box is over if the Sprint Goal becomes obsolete
  - Only the Product Owner has the authority to cancel the Sprint
  - All incomplete Product Backlog Items are re-estimated and put back on the Product Backlog

Before the First Sprint

- Before the first Sprint can begin, the Scrum team needs a **Product Backlog**, a prioritized (ordered 1, 2, 3, …) list of customer-centric features
  - The Product Backlog exists (and evolves) over the lifetime of the product; it is the product roadmap

				New Estimates at Sprint …					
Priority	Product backlog Item (PBI) Item	Details (wiki URL)	Initial Size Estimate	1	2	3	4	5	6
1	As a buyer, I want to place a book in a shopping cart (see UI sketches on wiki page)	…	5						
2	As a buyer, I want to remove a book in a shopping cart	…	2						
3	Improve transaction processing performance (see target performance metrics on wiki)	…	13						
4	Investigate solutions for speeding up credit card validation (see target performance metrics on wiki)	…	20						
5	Upgrade all servers to Apache 2.2.3	…	13						
6	Diagnose and fix the order processing script errors (bugzilla ID 14823)	…	3						
7	As a shopper, I want to create and save a wish list	…	40						
8	As a shopper, I want to to add or delete items on my wish list	…	20						

Sprint Planning

- Summary:
  - A meeting to prepare for the Sprint, typically divided into **two parts** (part one is **"what"** and part two is **"how"**)
  - To identify and prioritize the items to be implemented in this Sprint and decide how to implement the items
- Participants:
  - Part One: Product Owner, Team, ScrumMaster.
  - Part Two: Team, ScrumMaster, Product Owner (optional but should be reachable for questions)
- Duration:
  - Each part is **time-boxed** to one hour per week of Sprint

Sprint Planning Part One

- The Product Owner and Team review the high-priority items in the Product Backlog that the Product Owner is interested in implementing this Sprint
- Usually, these items will have been well-analyzed in a previous Sprint (during **Product Backlog Refinement**), so that at this meeting there are only minor last-minute clarifying questions
- The Product Owner and Team discuss the goals and context for these high-priority items on the Product Backlog, providing the Team with insight into the Product Owner's thinking
- The meeting focuses on understanding **what** the Product Owner wants and **why** they are needed
- The Team and the Product Owner may also devise the **Sprint Goal**
  - A summary statement of the Sprint objective (ideally has a cohesive theme)
- At the end of Part One the (always busy) Product Owner may leave although they must be available during Part Two of the meeting

Sprint Goal

- **Sprint goal** is a high-level summary of the goal the product owner would like to accomplish during a sprint, frequently elaborated through a specific set of product backlog items
  - Sprint goal is a short, one- or two-sentence, description of what the team plans to achieve during the sprint
  - It is written collaboratively by the team and the product owner
  - Examples of sprint goals for an e-Commerce application
    - Add, remove and update quantities for the shopping cart
    - Develop the checkout process: pay for an order, pick shipping, order gift wrapping
- Sprint goal can be defined through the following main steps:
  - Product owner presents the ordered backlog items to the team.
  - Team discusses and understands the work for this Sprint.
  - Team forecasts and commits on the items that can be done.
  - Team creates the Sprint Goal for this Sprint

Sprint Planning Part Two 1/2

- Team selects items from the product backlog which they commit to completing
- Team discusses how they will go about completing the selected items and create Sprint backlog
  - Tasks are identified for the selected items and the effort required for each task is estimated (1-16 hours)
    - High-level design is considered
    - Write acceptance test, design UI, write code, database, unit tests…
  - Collaboratively, not done alone by the ScrumMaster

As a vacation planner, I want to see photos of the hotels.

Code the middle tier (8 hours)
Code the user interface (4)
Write test fixtures (4)
Code the foo class (6)
Update performance tests (4)

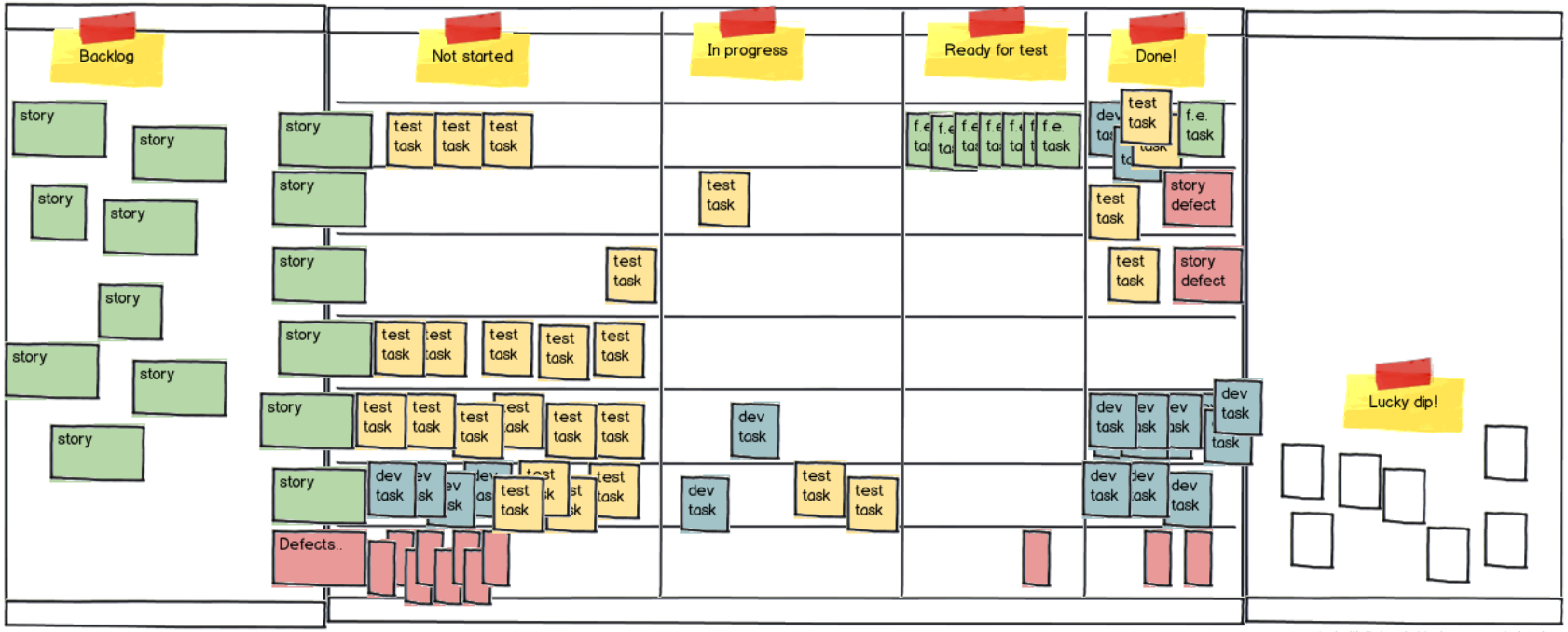
Sprint Planning Part Two 2/2

- The Team forecasts the amount of items they can complete by the end of the Sprint
  - Determine the available team **capacity** for the Sprint by using the **velocity** from the previous Sprints or by calculating the total team member's **available working hours** for the Sprint
- Team decomposes the Product Backlog items into fine-grained work (i.e., tasks) starting from the top of the Product Backlog and working down the list in order until the capacity is "full"
  - often starts with a design discussion at a whiteboard
  - this list of 'forecast" work to be done during the Sprint is called the **Sprint Backlog**
    - estimate the efforts required to do the tasks in hours
    - if the product backlog item is so small, the task can be the item itself
    - can have a Sprint Backlog in the form of a wall-sized task board (often called a **Scrum Board**)

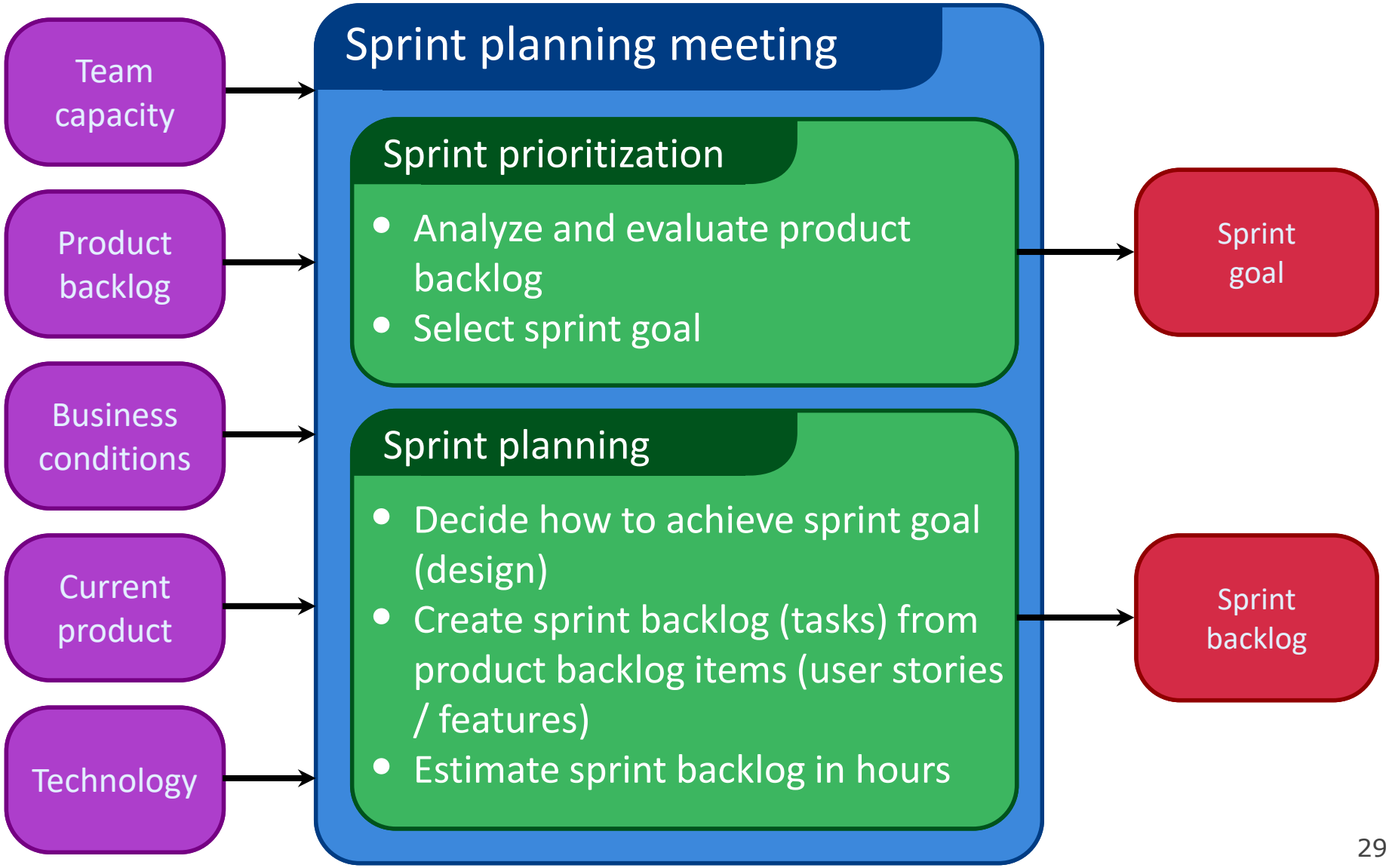
Create a Sprint Backlog

				New Estimates of Effort Remaining at end of Day...					
Product Backlog Item	**Sprint Task**	**Volunteer**	**Initial Estimate of Effort**	**1**	**2**	**3**	**4**	**5**	**6**
As a buyer, I want to place a book in a shopping cart	modify database		5						
	create webpage (UI)		8						
	create webpage (Javascript logic)		13						
	write automated acceptance tests		13						
	update buyer help webpage		3						
	. . .								
	merge DCP code and complete layer-level								

Sprint backlog Item (SBI)



Sprint Planning



Sprint Planning Meeting Explained

https://youtu.be/2A9rkiIcnVI (3:28)

Daily Scrum Meeting 1/2

- Summary:
  - Update and coordination between Team members
- Participants:
  - Team is required (everyone on the Team attends, anyone late pays a $1 fee); ScrumMaster is usually present but ensures Team holds daily Scrum meeting; Product Owner is optional
- Duration: (daily)
  - Maximum length of 15 minutes (stand-up)
- Three questions answered by each team member:
  - What did you do yesterday? (to help Team meet the Sprint Goal)
  - What will you do today? (to help Team meet the Sprint Goal)
  - What obstacles are in your way? (for meeting the Sprint Goal)
  - The intent of these questions is to emphasize **completions** of tasks, rather than effort spent
    - Having a clear definition of Done may help

Daily Scrum Meeting 2/2

- Not a status meeting to report to a manager
  - Generally recommended not to have managers or others in positions of perceived authority attend the Daily Scrum
- Not for problem solving
  - There is little or no in-depth discussion during the Daily Scrum
  - Someone makes note of the blocks, and the ScrumMaster is responsible to help Team members resolve them
  - If discussion is required, it takes place immediately after the Daily Scrum in one or more parallel follow-up meetings
    - Stakeholders can be in the follow-up meetings and offer to help with any blocks that are slowing the Team's progress
- Helps avoid other unnecessary meetings

The Daily Scrum Explained

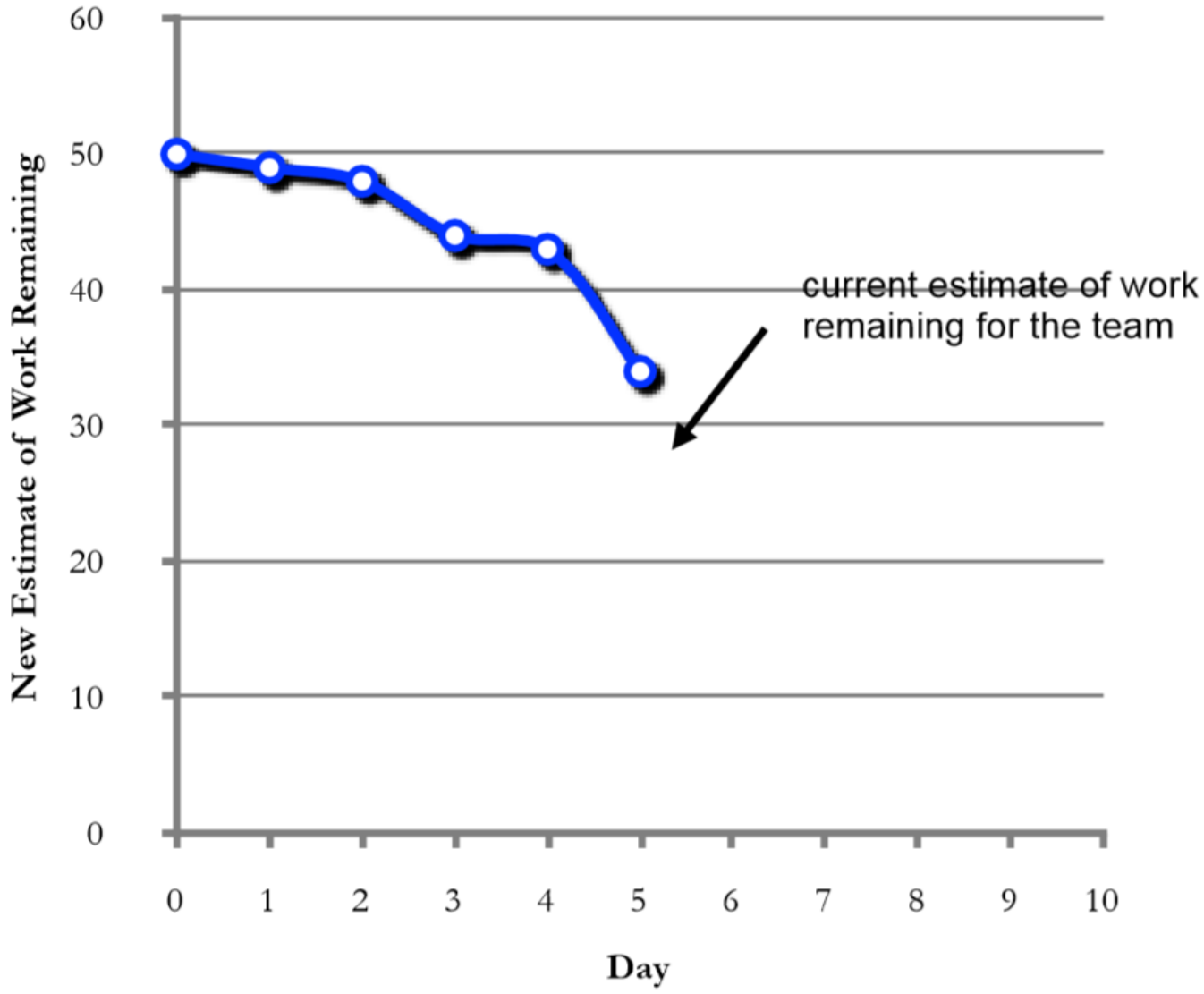
https://youtu.be/xcC0LmkzG9g (3:00)

Tracking Progress during the Sprint

- **Every day**, the Team members update their estimate of the effort remaining to complete their current work in the **Sprint Backlog**
  - add up the effort remaining for the Team as a whole and plot it on the Sprint **Burndown Chart**

				New Estimates of Effort Remaining at end of Day...					
Product Backlog Item	**Sprint Task**	**Volunteer**	**Initial Estimate of Effort**	**1**	**2**	**3**	**4**	**5**	**6**
As a buyer, I want to place a book in a shopping cart	modify database	Sanjay	5	4	3	0	0	0	
	create webpage (UI)	Jing	3	3	3	2	0	0	
	create webpage (Javascript logic)	Tracy & Sam	2	2	2	2	1	0	
	write automated acceptance tests	Sarah	5	5	5	5	5	0	
	update buyer help webpage	Sanjay & Jing	3	3	3	3	3	0	
	. . .								
Improve transaction processing performance	merge DCP code and complete layer-level tests		5	5	5	5	5	5	
	complete machine order for pRank		3	3	8	8	8	8	
	change DCP and reader to use pRank http API		5	5	5	5	5	5	

Sample Burndown Chart



Product Backlog Refinement
1/2

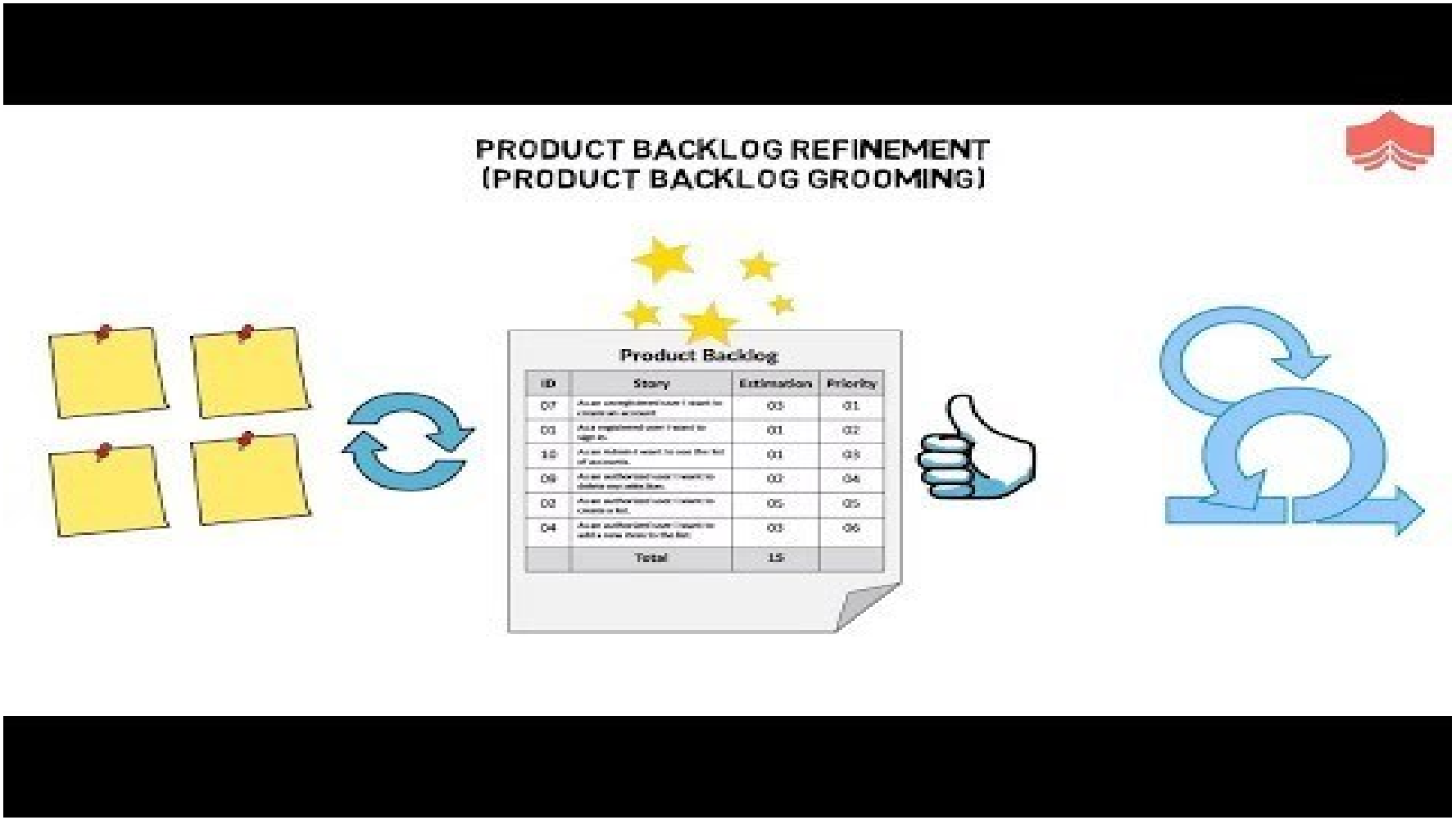
- Summary:
  - Split big items, analyze items, re-estimate, and re-prioritize, for future Sprints
    - The act of adding detail, estimates, and order to items in the Product Backlog
- Participants:
  - Team; Product Owner (will attend the entire activity if they are the expert who can help with the detailed refinement, otherwise they may attend only a subset to set direction or re-prioritize); ScrumMaster (will attend during initial sessions to coach the group to be effective, otherwise may not attend); Other stakeholders (who understand the requirements and can help the Team)
- Duration:
  - Usually, no more than 10% of the capacity of the Team for the Sprint, though it may be longer for "analysis heavy" items.
    - For example, in a two-week Sprint, perhaps one day is spent on refinement

Product Backlog Refinement
2/2

- The refinement activity includes detailed requirements analysis, splitting large items into smaller ones, estimation of new items, and re-estimation of existing items
- Can use a focused workshop near the middle or end of the Sprint, so that the Team and Product Owner and other stakeholders can dedicate themselves to this work without interruption
- The refinement activity is not for items selected for the current Sprint; it is for items for the future, most likely in the next one or two Sprints
- A sign that this refinement workshop is not being done (or not being done well) is that Sprint Planning involves significant questions, discovery, or confusion and feels incomplete

Product Backlog Refinement

- https://youtu.be/pSguy2FuC2c (3:31)



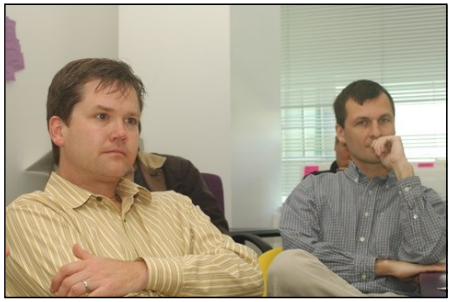
The Sprint Review 1/3

- Summary:
  - Inspection and adaption related to the product increment of functionality and adapt the Product Backlog if needed
  - Informal meeting in which Team presents what it accomplished during the sprint
  - Typically takes the form of a demo of new features (**live software**) or underlying architecture (**no slideware**)
- Participants:
  - Team, Product Owner, ScrumMaster. Other stakeholders as appropriate, invited by the Product Owner
- Duration:
  - Time-boxed to one hour per week of Sprint

The Sprint Review 2/3

- Story points count towards velocity once the whole work item has been completed and it's in a potentially shippable state;
  - no points are awarded for work items that are partially completed
- The Review is often mislabeled the "demo" but that does not capture the real intent of this meeting
  - The Sprint Review is an inspect and adapt activity for the **product**
  - It is a time
    - for the Product Owner to learn what is going on with the product and with the Team (that is, a review of the Sprint); and
    - for the Team to learn what is going on with the Product Owner and market
    - a critical element of the Review is an **in-depth conversation** between the Team and Product Owner to learn the situation, to get advice, and so forth

Aim to spend no more than 30 minutes preparing for Sprint Review, otherwise it suggests something is wrong



The Sprint Review 3/3

- The result of the Sprint Review is
  - a revised Product Backlog that defines the probable Product Backlog items for the next Sprint; and
  - the total story points of the "Done" items in this Sprint and the average velocity
- The Sprint Review may include the following elements
  - The Product Owner explains what Product Backlog items have been "Done" and what has not been "Done"
  - The Development Team discusses what went well during the Sprint, what problems it ran into, and how those problems were solved
  - The Development Team demonstrates the work that it has "Done" and answers questions about the Increment
  - The Product Owner discusses the Product Backlog as it stands. He or she projects likely target and delivery dates based on progress to date (if needed)
  - The entire group collaborates on what to do next, so that the Sprint Review provides valuable input to subsequent Sprint Planning
  - Review of how the marketplace or potential use of the product might have changed what is the most valuable thing to do next
  - Review of the timeline, budget, potential capabilities, and marketplace for the next anticipated releases of functionality or capability of the product

The Sprint Review

https://youtu.be/dctXACpVC1U (1:59)

The Sprint Retrospective 1/2

- Summary:
  - Inspection and adaption related to the **process** and **environment**
  - Periodically take a look at what is and is not working, and agree on changes to try
  - Typically follows the Sprint Review
- Participants:
  - Team, ScrumMaster, Product Owner (optional). Other stakeholders may be invited by the Team, but are not otherwise allowed to attend
- Duration:
  - Timeboxed to 45 minutes per week of Sprint
  - Typically 15–30 minutes

The Sprint Retrospective 2/2

- The Scrum Master ensures that the meeting is **positive** and **productive**
  - ScrumMaster can act as an effective facilitator for the Retrospective
  - Do not only focus on problems; shall also focus on positives or strengths
    - Ex: each team member can tell one thing "good" and one thing "to be improved" or have a ordered list of what "went well" and "could be better"

- The purpose of the Sprint Retrospective is to:
  - Inspect how the last Sprint went with regards to **people**, **relationships**, **process**, and **tools**
  - Identify and order the major items that went well and potential improvements (what went well and what could be better)
  - Create a plan for implementing improvements to the way the Scrum Team does its work
- By the end of the Sprint Retrospective, the Scrum Team should have identified improvements that it will implement in the next Sprint

How to run the Sprint Retrospective

https://youtu.be/n_iu8kuA0XE (3:00)

The Sprint Retrospective

https://youtu.be/gxuRYcCwxeI (2:25)

Starting the next Sprint 1/2

- Following the Sprint Review, the Product Owner may update the Product Backlog with any new insight
  - adding new Items, removing obsolete ones, or revising existing ones
  - the Product Owner is responsible for ensuring that these changes are reflected in the Product Backlog

				New Estimates at Sprint ...				
Priority	**Item**	**Details (wiki URL)**	**Initial Estimate**	**1**	**2**	**3**	**4**	**5**
1	As a buyer, I want to place a book in a shopping cart (see UI sketches on wiki page)	...	5	0	0	0		
2	As a buyer, I want to remove a book in a shopping cart	...	2	0	0	0		
3	Improve transaction processing performance (see target performance metrics on wiki)	...	13	13	0	0		
4	Investigate solutions for speeding up credit card validation (see target performance metrics on wiki)	...	20	20	20	0		
5	Upgrade all servers to Apache 2.2.3	...	13	13	13	13		
6	Diagnose and fix the order processing script errors (bugzilla ID 14823)	...	3	3	3	3		
7	As a shopper, I want to create and save a wish list	...	40	40	40	40		
8	As a shopper, I want to to add or delete items on my wish list	...	20	20	20	20		
	...		...	...				
			537	**580**	**570**	**500**		

Starting the next Sprint 2/2

- There is no down time between Sprints
  - Teams normally go from a Sprint Retrospective one afternoon into the next Sprint Planning the following morning (or after the weekend)

- One agile principles is "sustainable pace"
  - Teams can continue this cycle indefinitely only **by working regular hours at a reasonable level**
  - Productivity grows over time through the evolution of the Team's practices, and the removal of impediments to the Team's productivity, not through overwork or the compromise of quality

Scrum Framework

Roles

- Product owner
- Scrum Master
- Team

Events

- Sprint planning
- Daily scrum meeting
- Backlog refinement
- Sprint review
- Sprint retrospective

Artifacts

- Product backlog
- Sprint backlog
- Product Increment

Product Backlog 1/2

This is the product backlog

- The requirements
- A list of all desired work on project
- Can be expressed as a list of user stories along with "story points" (size of the item)
- Each item has a value (ROI) to users or customers of the product
- Prioritized by the product owner
- Reprioritized at start of each sprint

Product Backlog 2/2

- The Product Backlog is an ordered list of everything that is known to be needed in the product
  - It lists all features, functions, requirements, enhancements, and fixes that constitute the changes to be made to the product in future releases
- The Product Backlog does not contain "user stories"; it simply contains items. Those items can be expressed as user stories, use cases, or any other requirements approach that the group finds useful
- A Product Backlog is never complete. It is **dynamic** and **constantly changes** (**living artifact**)
- Product Backlog items can be updated at any time by the Product Owner or at the Product Owner's preference
- The Development Team is responsible for all estimates
  - The Product Owner may influence the Development Team by helping it understand and select trade-offs, but the people who will perform the work make the final estimate

Good Product Backlog

- A good Product Backlog is **DEEP**
  - **Detailed appropriately**
    - The top priority items are more fine-grained and detailed than the lower priority items (coarse grained)
  - **Estimated**
    - The items for the current release need to have estimates, and furthermore, should be considered for re-estimation each Sprint as everyone learns and new information arises
  - **Emergent**
    - In response to learning and variability, the Product Backlog is regularly refined
  - **Prioritized**
    - The items are prioritized or ordered in a 1-N order
    - The highest-priority items should deliver the most bang for your buck: lots of bang (business value) for low buck (cost).
    - Another motivation to increase the priority of an item is to tackle high risks early, before the risks attack you

Note on Product Backlog

- **Major engineering improvements** that consume much time and money should be in the Product Backlog, since they may be an optional business investment, ultimately to be made by the business-oriented Product Owner
  - The improvement goal is created in the previous Sprint's Retrospective

- Note that in Scrum, the Team has independent authority of how many items from the Product Backlog they decide to take into a Sprint, so they are **independently free** to take on **minor engineering improvement work** as they can be considered part of the normal cost of doing business and what is required for a developer to do their job properly

- One of the myths about Scrum is that it prevents you from writing detailed specifications; in reality, it is up to the Product Owner and Team to decide how much detail is required, and this will vary from one backlog item to the next, depending on the insight of the Team, and other factors

User Stories

- The format that commonly used to express a user story is
  - **Who** (user role) – Is this a customer, employee, admin, etc.?
  - **What** (goal) – What functionality must be achieved/developed?
  - **Why** (reason) – Why does user want to accomplish this goal?

As a [user role], I want to [goal], so that I can [reason].

As a *<user>* I want *<function>* so that *<value>*

- Example:
  - "As a user, I want to log in, so that I can access subscriber content."

- **Story points (or simply points)**: Rating of effort needed to implement this story
  - in terms of relative size (factoring in effort, complexity, uncertainty)
  - common scales: 1-10, shirt sizes (XS, S, M, L, XL), etc.

Sample Product Backlog

Backlog item	Estimate
Allow a guest to make a reservation	3 (story points)
As a guest, I want to cancel a reservation.	5
As a guest, I want to change the dates of a reservation.	3
As a hotel employee, I can run RevPAR reports (revenue-per-available-room)	8
Improve exception handling	8
…	30
…	50

A Good User Story

- A set of criteria, called **INVEST**, is used to assess the quality of a user story
  - **Independent:** stories should be as independent as possible (in terms of implementation order)
  - **Negotiable:** a story is not a contract. It is the result of collaborative negotiation (conversation) between the customer and the team
  - **Valuable:** the story has value to the "user" in the user story
  - **Estimable:** a story has to be able to be estimated (to a decent approximation) or sized so it can be properly prioritized
  - **Small:** stories are small enough to fit in a single iteration
  - **Testable:** Every story needs to be testable in order to be "done" (met the acceptance criteria)

Story Point Estimation (Planning Poker)

https://youtu.be/TxSzo3lwwWQ (3:14)

Sprint Backlog 1/2

- The list of work to be done during the Sprint is called the Sprint Backlog
  - Each item in the Sprint backlog should be small enough so that it can be "done" within one Sprint
    - A common guideline is that an item is estimated small enough to complete within one fourth or less of a Sprint by the whole Team
  - To ensure continuous improvement, it includes at least one high priority process improvement identified in the previous Retrospective meeting
- Estimated work remaining is updated daily
- Any team member can add, delete change sprint backlog
- Work for the sprint emerges (dynamic)
- If work is unclear, define a sprint backlog item with a larger amount of time and break it down later
- Update work remaining as more becomes known

Sample Sprint Backlog 1/2

Tasks	Mon	Tue	Wed	Thu	Fri
Code the user interface	8	4	8	update	
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12			delete	
Write the Foo class	8	8	8	8	8
Add error logging		new add	8	4	

Sample Sprint Backlog 2/2

Sprint 1

01/11/2004

19 days work in this sprint

	Backlog Item	Backlog Item	Owner	Estimate	Sprint Day	1 Mo	2 Tu	3 We	4 Th	5 Fr	6 Sa	7 Su
					Hours remaining	152	152	152	152	152	152	152
1	Minor	Remove user kludge in .dpr file	BC	8		8	8	8	8	8	8	8
2	Minor	Remove cMap/cMenu/cMenuSize from disciplines.pas	BC	8		8	8	8	8	8	8	8
3	Minor	Create "Legacy" discipline node with old civils and E&I content	BC	8		8	8	8	8	8	8	8
4	Major	Augment each tbl operation to support network operation	BC	80		80	80	80	80	80	80	80
5	Major	Extend Engineering Design estimate items to include summaries	BC	16		16	16	16	16	16	16	16
6	Super	Supervision/Guidance	CAM	32		32	32	32	32	32	32	32

Sprint 1

01/11/2004

19 days work in this sprint

	Backlog Item	Backlog Item	Owner	Estimate	Sprint Day	1 Mo	2 Tu	3 We	4 Th	5 Fr	6 Sa	7 Su
					Hours remaining	152	150	140	130	118	118	118
1	Minor	Remove user kludge in .dpr file	BC	8		8	8	4	2	0		
2	Minor	Remove cMap/cMenu/cMenuSize from disciplines.pas	BC	8		8	8	4	0			
3	Minor	Create "Legacy" discipline node with old civils and E&I content	BC	8		8	8	8	6	0		
4	Major	Augment each tbl operation to support network operation	BC	80		80	80	80	80	78	78	78
5	Major	Extend Engineering Design estimate items to include summaries	BC	16		16	16	16	16	16	16	16
6	Super	Supervision/Guidance	CAM	32		32	30	28	26	24	24	24

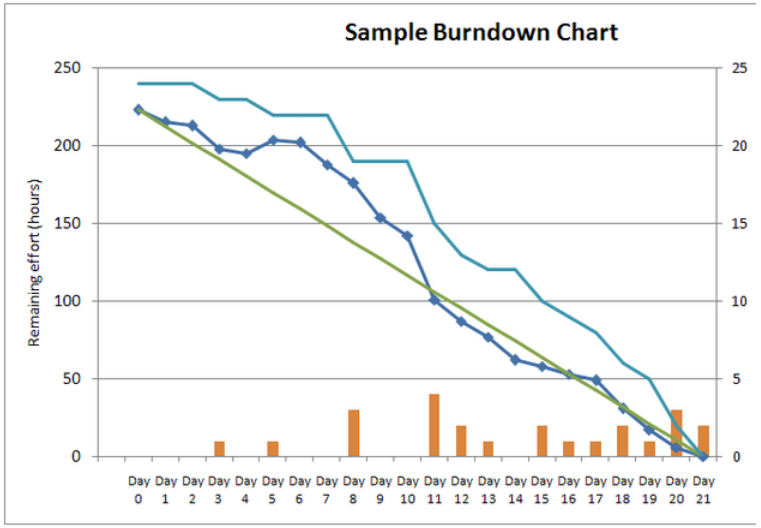
Sprint Backlog 2/2

- Individuals sign up for work of their own choosing
  - Work is never assigned
- In Sprint Planning, it is not necessary – nor appropriate – for people to volunteer for all the tasks "they can do best"
  - It is better to only volunteer for one task at a time, when it is time to pick up a new task, and to consider choosing tasks that will on purpose involve learning (perhaps by pair work with a specialist)
- Only the Development Team can change its Sprint Backlog during a Sprint
  - The Sprint Backlog is a highly visible, real-time picture of the work that the Development Team plans to accomplish during the Sprint, and it belongs solely to the Development Team

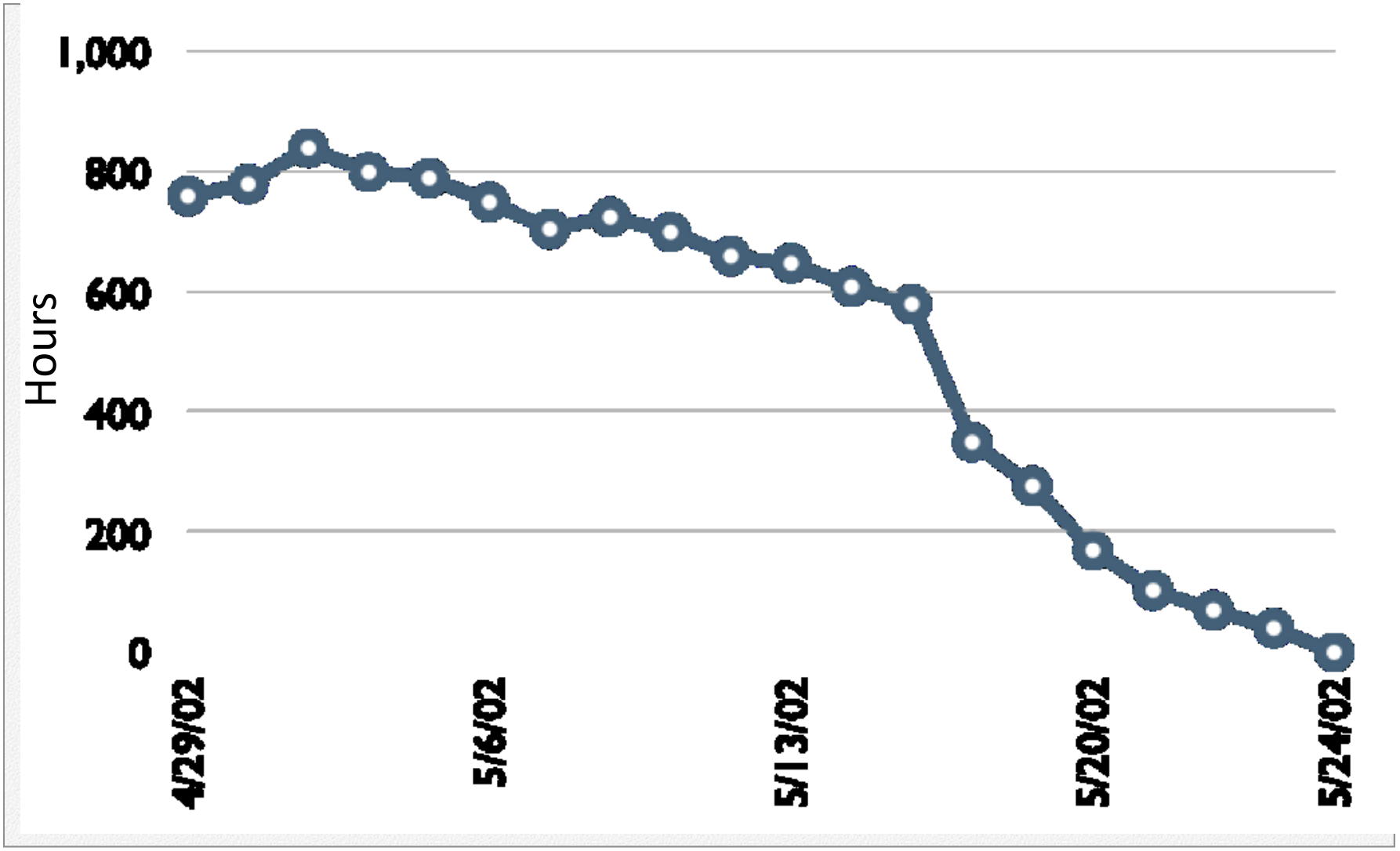
Sprint Burndown Chart

- A display of what work has been completed and what is left to complete
  - one for each developer or work item
  - **updated every day**
  - (make best guess about hours/points completed each day)

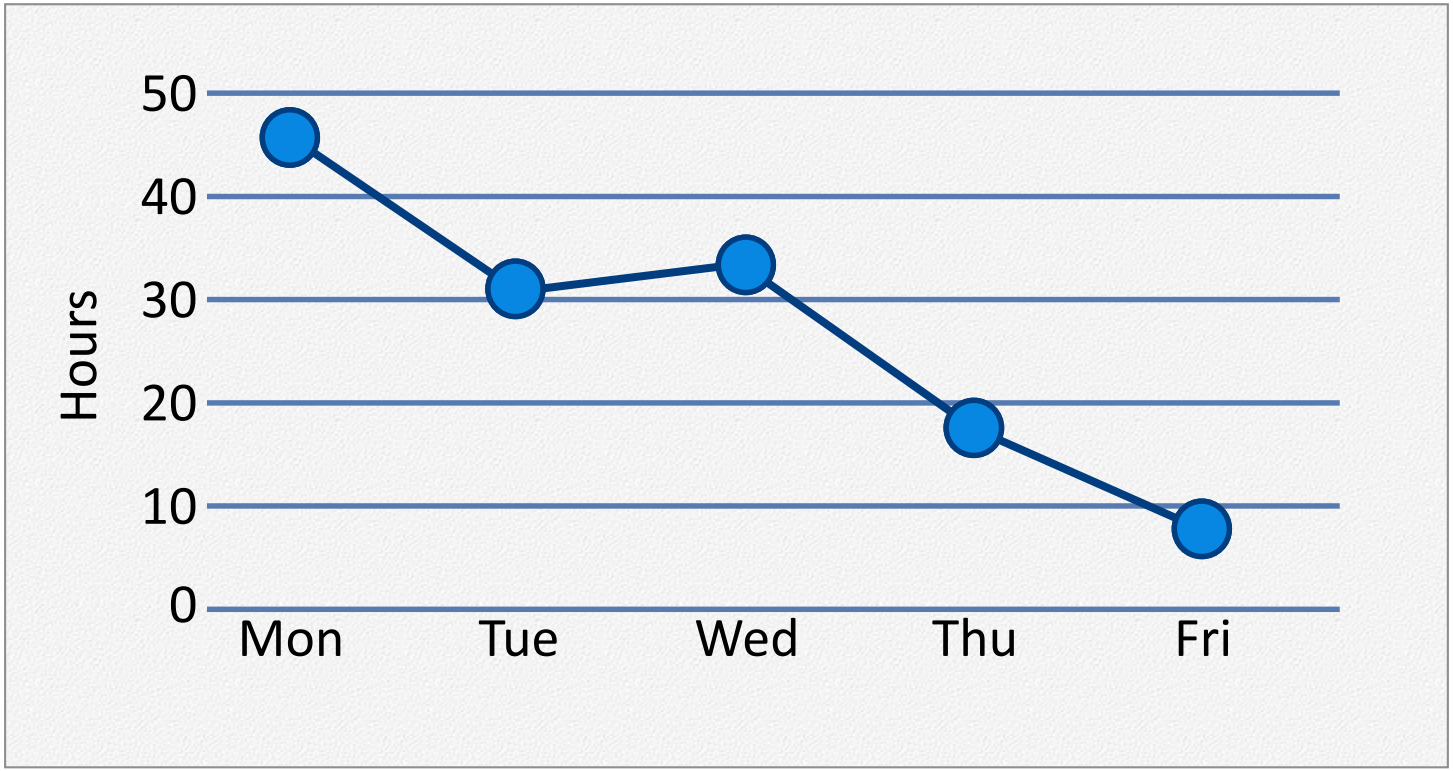
- *Variation:* Release burndown chart
  - shows overall progress
  - updated at end of each sprint



Sample Burndown Chart

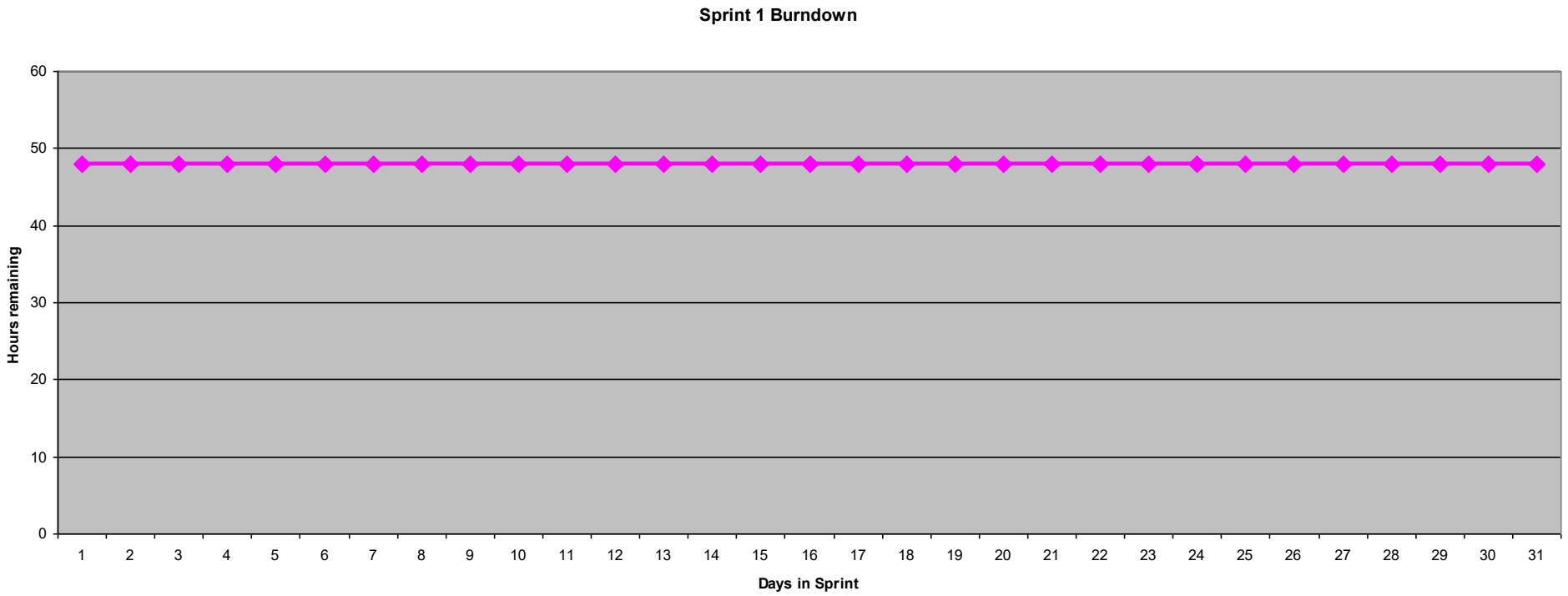


Tasks	Mon	Tue	Wed	Thu	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	7	
Test the middle tier	8	16	16	11	8
Write online help	12				



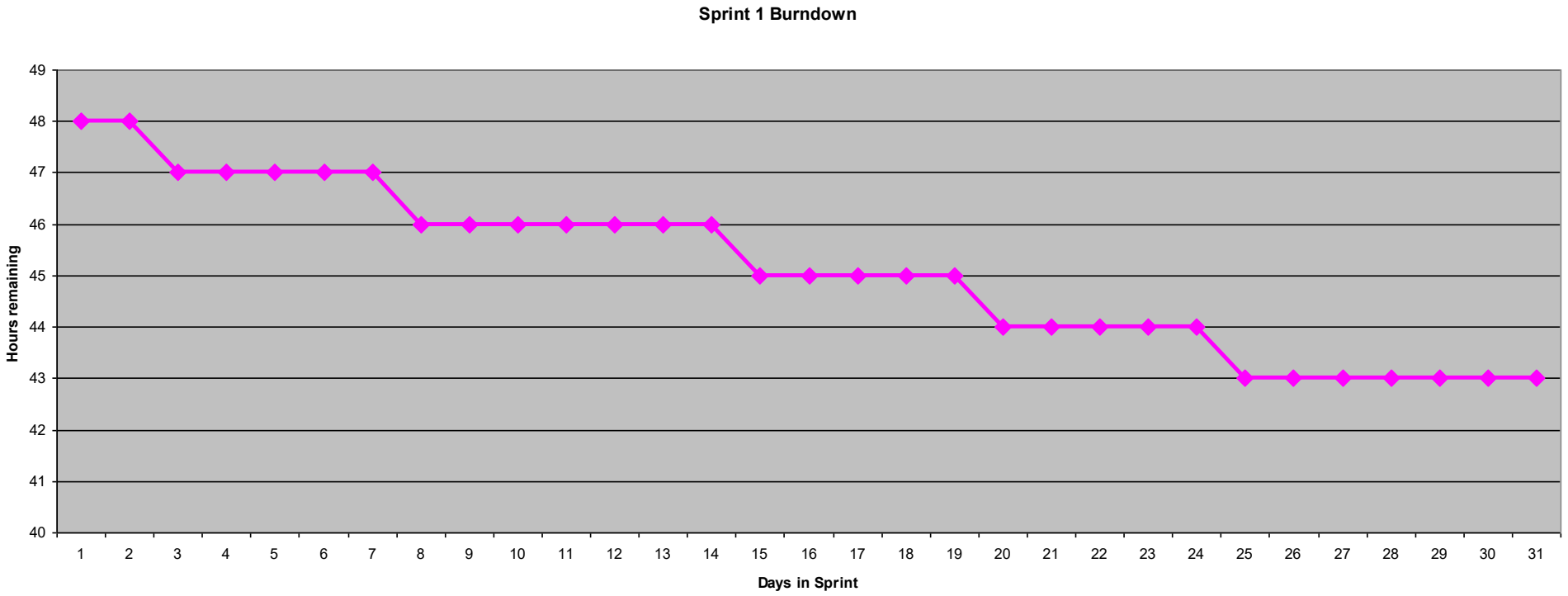
Burndown Example 1

No work being performed



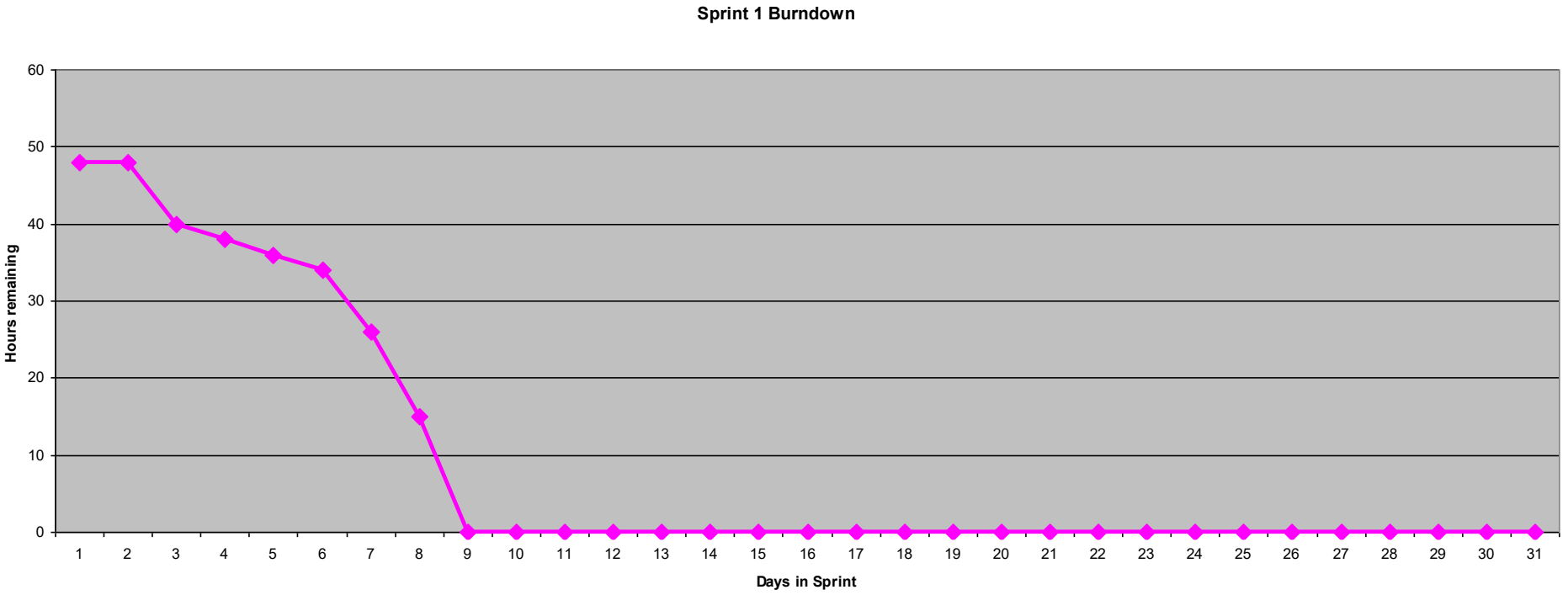
Burndown Example 2

Work being performed, but not fast enough



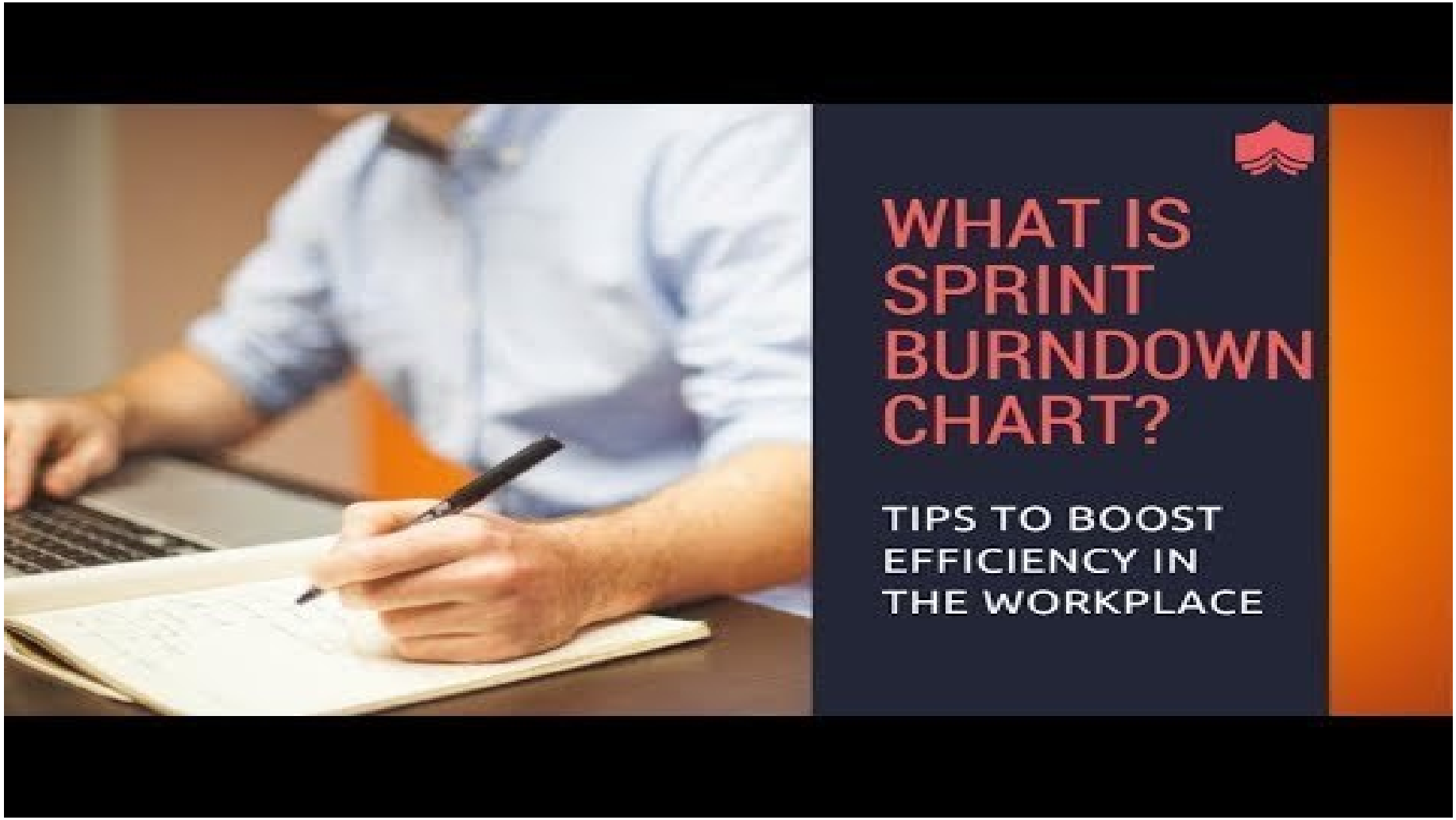
Burndown Example 3

Work being performed, but too fast!



What Is Sprint Burndown Chart?

https://youtu.be/rCY0XrSef1A (3:24)



How to use The Sprint Burndown

https://youtu.be/GokN-50Jt4A (2:48)

Potentially shippable Increment

- The Increment is the sum of all the Product Backlog items completed during a Sprint and the value of the increments of all previous Sprints
  - At the end of a Sprint, the new Increment must be "Done," which means it **must** be in **useable condition** and meet the Scrum Team's definition of "**Done**"

- **Potentially shippable software** is a feature(s) that can be released, with limited notice, to end users at the product owner's discretion
  - Ideally, potentially shippable is equivalent to the Definition of Done (DoD)

Definition of Done (DoD)

- When a Product Backlog item or an Increment is described as "Done", everyone must understand what "Done" means
  - **All activities** that are needed in order to ship the product should be included in the definition of **Potentially Shippable** and therefore **should be done during the Sprint**
    - **A checklist**

- The Product Owner and Team need to agree on a Definition of Done
  - Have a shared understanding

- The Team will plan their Sprint work according to this Definition of Done
  - A reporting mechanism used by the Team
  - DoD is used to assess when work is complete on the product Increment

- The DoD changes over time based on team experience maturity and can vary among different teams

Non-Static DoD

- Teams with little experience in Scrum should start with relatively simple definitions of done
- As teams gain experience their Definition of Done matures and they find value in the Definition of Ready
- As teams continue to mature they realize "Ready" is about starting well and that "Done" is about ending well

- Teams should, "Discuss all of the obstacles which stop them from delivering this each iteration/sprint". Common **root causes** for impediments include:
  - Team does not have the skillset to incorporate activities into the definition of done for a sprint or for a feature
  - Team does not have the right set of tools. (Example: continuous integration environment, automated build, servers etc.)
  - Team members are executing their sprint in mini-waterfalls
  - Aha! Here's an opportunity to be more cross-functional and share responsibilities across functional silos

Components of DoD

- Definition of Done has Two Components
  - **Policies** about how to test and validate deliverables
    - Examples
      - All defects found when testing a Story must be fixed immediately
      - Acceptance tests must be satisfied
      - Decide which environment to use: QA? Dev? Either?
      - Unit tests must be satisfied
      - …
  - **Checklists** of required tasks that reflect organizational standards
    - Examples
      - Code is checked in to repository
      - Build from repository is successful
      - Deployment process is successful
      - Code has been reviewed
      - Unit tests have been developed
      - Acceptance tests are automated
      - …

Examples of DoD

We must meet the DoD to ensure product quality and reduce rework

- The most common use of DoD is on the delivery team level which means the Product Owner reviewed and accepted the **user story**
  - Unit tests passed
  - Code reviewed
  - Acceptance criteria met
  - Functional tests passed
  - Non-Functional requirements met
  - Product Owner accepts the User Story
- Done on **feature level** may mean the feature qualifies to add to a release
  - Acceptance criteria met
  - Integrated into a clean build
  - Promoted to higher level environment
  - Automated regression tests pass
  - Feature level functional tests passed
  - Non-Functional requirements met
  - Meets compliance requirements
  - Functionality documented in necessary user documentation

Levels of DoD

- In reality, many teams are still working towards a potentially shippable state. Such teams may have a different DoD at various levels:
  - DoD for a **feature** (story or product backlog item)
  - DoD for a **sprint** (collection of features developed within a sprint)
  - DoD for a **release** (potentially shippable state)

- There are various factors which influence whether a given **activity** belongs in the DoD for a feature, a sprint or a release
  - Ultimately, the decision rests on the answer to the following three questions:
    - Can we do this activity for **each feature**? If not, then
    - Can we do this activity for **each sprint**? If not, then
    - We have to do this activity for our **release**!

Definition of Ready (DoR)

- The "definition of ready" provides clarity to a product owner (or product owner team) on what it means to create ready product backlog items
  - a checklist to determine whether a story is Ready (in an actionable state)
- Examples of Definition of Ready for a user story
  - All stories must meet the INVEST criteria
  - A clearly defined user
  - An estimate
  - Acceptance criteria with examples
  - The team should understand how to provide a demo of the features
- Examples of Definition of Ready for a sprint
  - Prioritized sprint backlog
  - All users stories meet the definition of Ready
  - All team members have calculated their individual capacity for the sprint

Acceptance Criteria

- **Acceptance Criteria** are a set of statements, each with a clear pass/fail result, that specify both functional and non-functional requirements

- The **Given/When/Then** format is helpful way to specify criteria
  - *Given some precondition When I do some action Then I expect some result*

- If the above format is difficult to construct (sometimes for system level user stories), in that case you may use a verification checklist
- Acceptance criteria are about specific characteristics of **a Product Backlog Item**
  - DoD is about **policies** that apply **across Product Backlog Items**. It is not just Acceptance Criteria

Summary of DoD and DoR

- You must have a Definition of Done
  - Start small: Shippable Quality
  - Evolve over time to reflect customer/product needs
  - Adjust as your mature your automation and CI/CD practices

- You should have an Agreement of Ready
  - Use it to promote healthy conversations
  - Don’t treat it as rigid or static

- As a PO, the team and I need clear definitions of **READY and DONE** so the team can complete the backlog quickly and effectively

Managing Releases

- For a new product, or an existing product just adopting Scrum, there is the need to do **initial Product Backlog refinement** before the **first Sprint**, where the Product Owner and Team shape a proper Scrum Product Backlog
  - This could take a few days or a week, and involves a workshop (sometimes called **Initial Product Backlog Creation** or **Release Planning**), some detailed requirements analysis, and estimation of all the items identified for the **first release**
- There should be no needs for any special or extensive release planning for the next release
  - Sprints continue until the Product Owner decides the product is ready for release
  - During an initial Product Backlog refinement workshop and during the continuous backlog refinement each Sprint, the Team and Product Owner will do **release planning**, refining the estimates, priorities, and content as they learn

Theory of Scrum

- Scrum is founded on empirical process control theory (經驗導向的流程控制理論), or **empiricism (經驗主義)**
- Empiricism asserts that **knowledge comes from experience** and **making decisions based on what is known**
- Scrum employs an iterative, incremental approach to optimize predictability and control risk
- Three pillars uphold every implementation of empirical process control: transparency, inspection, and adaptation.

Scrum Three Pillars

- **Transparency**
  - Significant aspects of the process must be **visible** to those responsible for the outcome
  - Transparency requires those aspects be defined by a common standard so observers share a **common understanding** of what is being seen
  - Examples:
    - common language referring to process, common definition of Done
- **Inspection**
  - Users must **frequently inspect** Scrum artifacts and progress toward a Sprint Goal to detect undesirable variances
- **Adaptation**
  - The process or the material being processed must be adjusted as soon as possible to minimize further deviation
- Four events for inspection and adaption
  - Sprint Planning, Daily scrum meeting (adapt for next day),
  - Sprint review meeting (adapt for next sprint), Retrospective meeting

Scrum Values

- When the values of **commitment**, **courage**, **focus**, **openness** and **respect** are embodied and lived by the Scrum Team, the **Scrum pillars** of transparency, inspection, and adaptation come to life and build trust for everyone

- Successful use of Scrum depends on people becoming more proficient in living these five values
  - People personally commit to achieving the goals of the Scrum Team
  - The Scrum Team members have courage to do the right thing and work on tough problems
  - Everyone focuses on the work of the Sprint and the goals of the Scrum Team
  - The Scrum Team and its stakeholders agree to be open about all the work and the challenges with performing the work
  - Scrum Team members respect each other to be capable, independent people

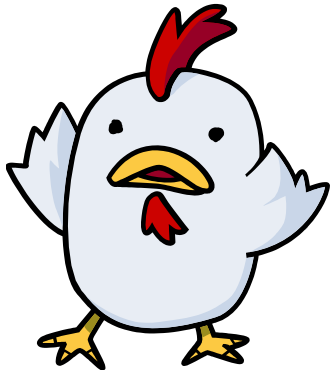
- **People, Not Process**

"Pigs" and "Chickens"

- **Pig**: Team member committed to success of project
- **Chicken**: Not a pig; interested but not committed

A pig and a chicken are walking down a road. The chicken looks at the pig and says, "Hey, why don't we open a restaurant?" The pig looks back at the chicken and says, "Good idea, what do you want to call it?" The chicken thinks about it and says, "Why don't we call it 'Ham and Eggs'?" "I don't think so," says the pig, "I'd be committed but you'd only be involved."

commit

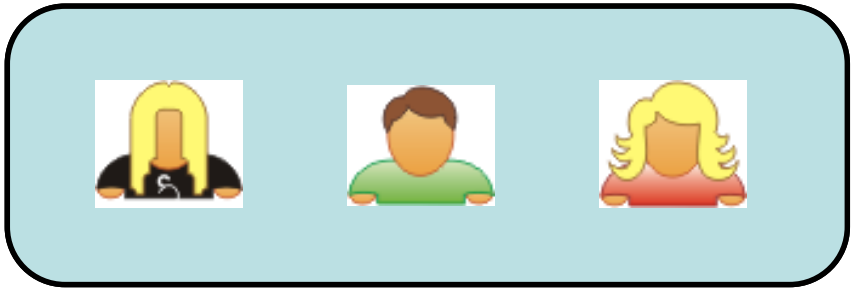


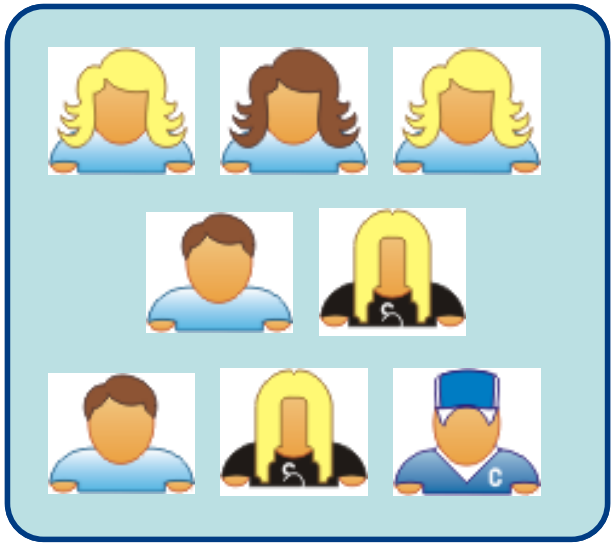
involve

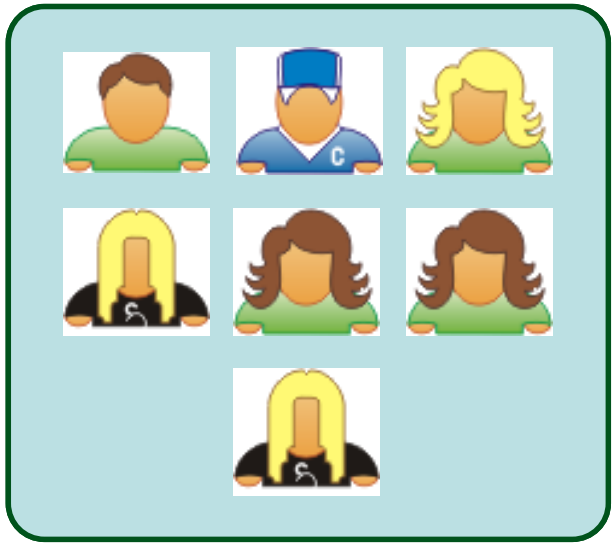
Scalability

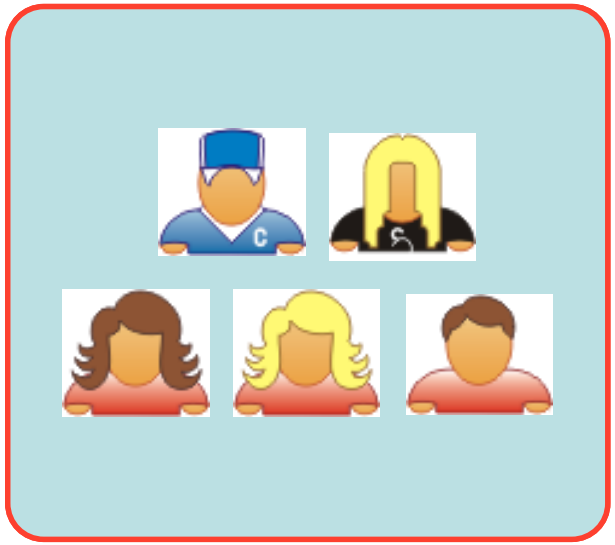
- Scrum has been used on multiple 500+ person projects
  - Scalability comes from **teams of teams**
- Factors in scaling
  - Type of application
  - Team size
  - Team dispersion
  - Project duration
- **Scrum of Scrums**
  - A technique to scale Scrum up to large groups (over a dozen people), consisting of dividing the groups into Agile teams of 5-10
  - Each daily Scrum includes a designated member as "ambassador" in order to participate in a daily meeting with ambassadors from other teams, called Scrum of Scrums
    - Depending on the context, ambassadors may be technical contributors, or each team's ScrumMaster, or even managers of each team

Scaling through the Scrum of scrums

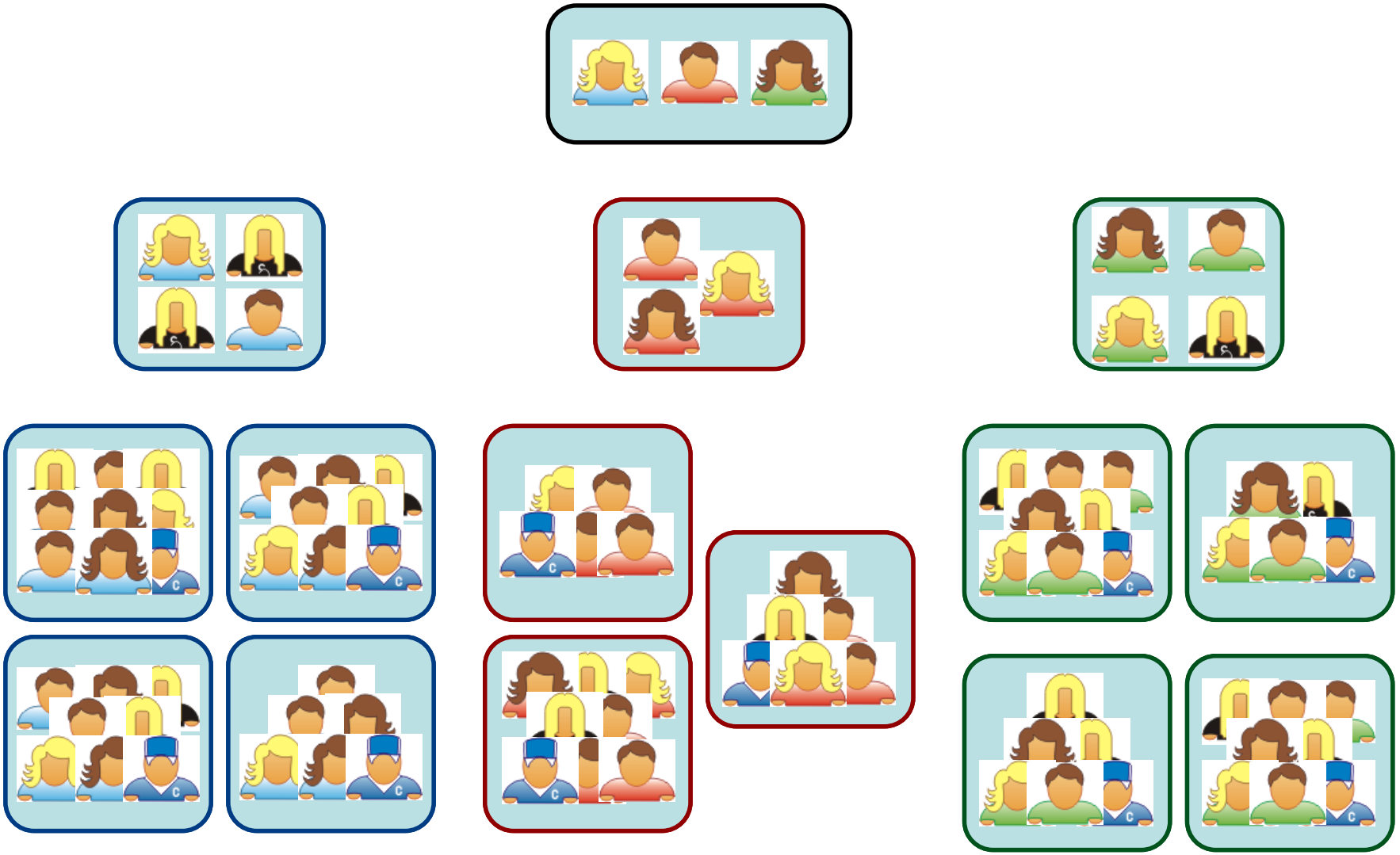








Scrum of scrums of scrums



Summary

- Scrum is not only a concrete set of practices – rather, and more importantly, it is a **framework** that provides transparency, and a mechanism that allows "inspect and adapt"

- Scrum does not solve the problems of development
  - **It makes them painfully visible**, and provides a framework for people to explore ways to resolve problems in short cycles and with small improvement experiments

- For successfully implementing Scrum
  - The characteristics and mindset of Scrum team are essential
  - Software engineering practices and skills are very important
  - The culture and support of the organizations are critical
  - Continuously inspect your Scrum process, diagnoses, adapt and improve
    - Scrum is simple to understand, but difficult to master

References

- Scrum Primer https://scrumprimer.org/
- Scrum Guide https://scrumguides.org/
- Redistributable slide from Mountain Goat LLC (An Introduction to Scrum)
- Agile Alliance https://www.agilealliance.org/
- Scrum Alliance https://www.scrumalliance.org/
- YouTube Video related to Scrum (Knowledgehut)
- Scrum related articles over Internet