Agile Project Management using SCRUM

Alfredo E. Flores Jr.
Facilitator

JRU and September 2017

Facilitator

- **Alfredo E. Flores Jr.**
  - Certified Project Management Profession
  - Certified Scrum Master
  - Masters in Information Management
  - Agile Coach and Facilitator
  - Lean Practitioner
  - Experienced IT professional with exposure to global organizations and various industries like Oil & Energy, Healthcare and Pharmaceuticals, Corporate and Finance Shared Services, and Training and Consulting.

Topic	Size	Est. Duration	Start
History	S	10	9:30
What is Agile?	S	10	
Agile vs Traditional	S	5	
Scrum Overview	M	20	
Writing ‘Agile’ Requirements	L	45	10:30
Lunch Break		45	
Estimation	M	20	
Planning	L	45	2:00
Task breakdown	M	15	
Tracking Progress	M	10	
Timeboxed Meetings	M	15	
Wrap up			3:00

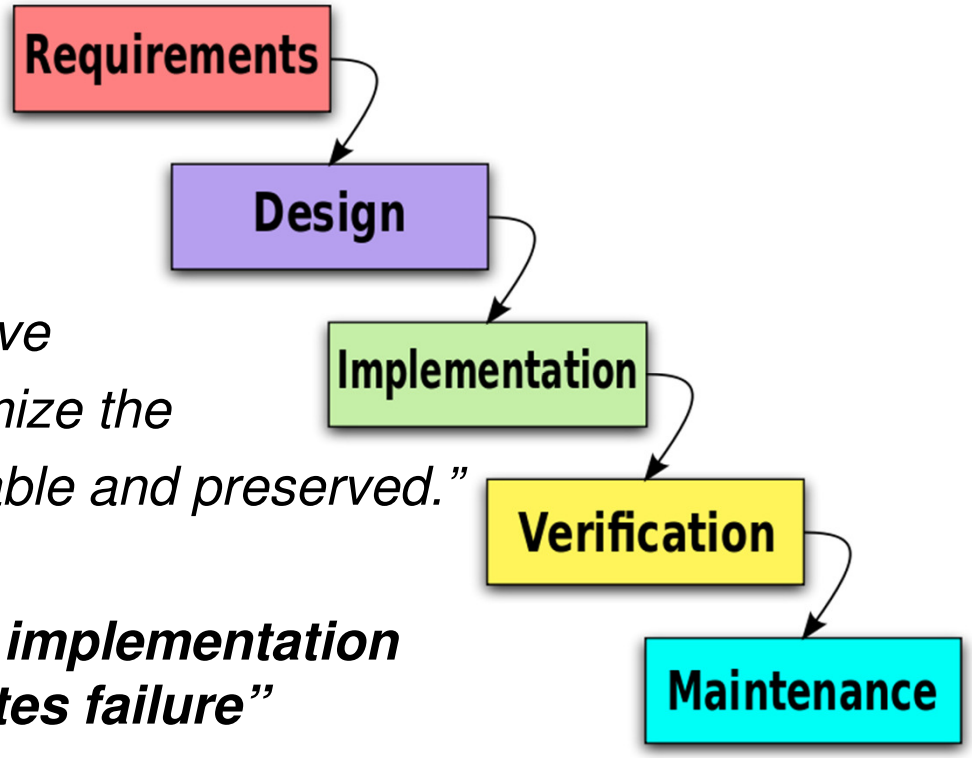
History of Agile

Before Agile

Waterfall

1970 article

by Winston W. Royce. Royce



Page 1: *"What we have is an effective fallback position that tends to maximize the extent of early work that is salvageable and preserved."*

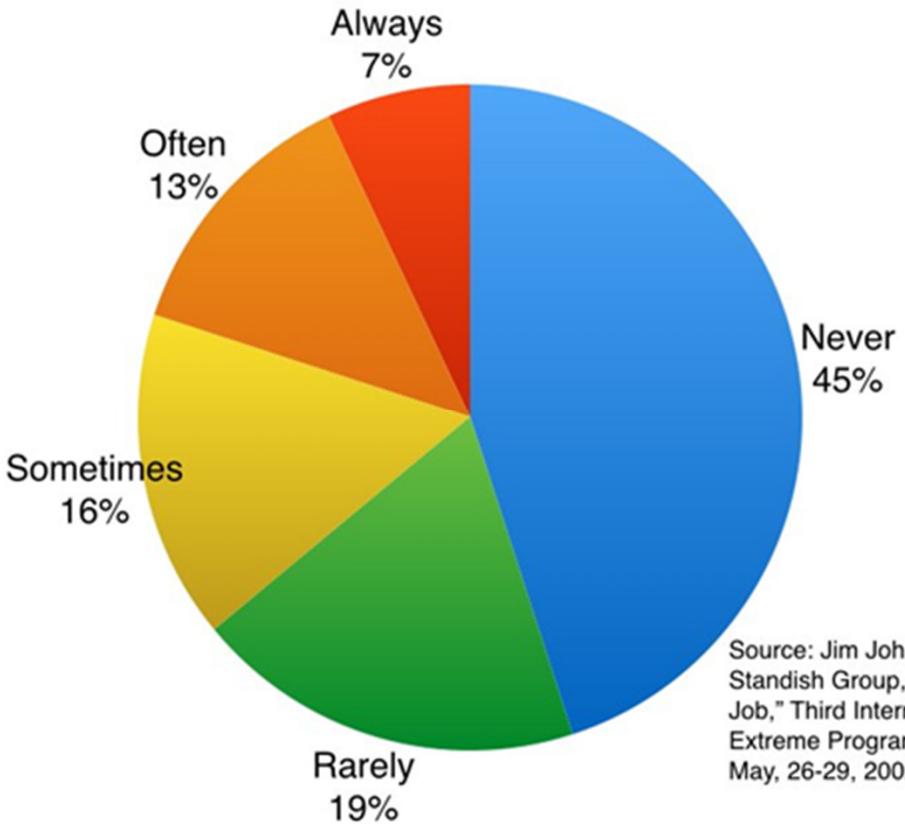
…but on Page 2:

"I believe in this concept, but the implementation described above is risky and invites failure"

Traditional "waterfall" development depends on a **perfect understanding** of the product requirements at the outset and minimal errors executing each phase.

Source: Royce, Winston (1970), *Managing the Development of Large Software Systems*

Project Management Crisis



Source: Jim Johnson, Chairman of The Standish Group, Keynote "ROI, It's Your Job," Third International Conference on Extreme Programming, Alghero, Italy, May, 26-29, 2002.

Standish Group Report

84% of projects were either challenged or critical failure

64% of features are never and rarely used

It's not the people, it's the process. There should be a fundamental shift on the way we do things.

What is Agile?

What is Agile?

Agile **is a set of values and principles.** It is **not a process, methodology, practices, or tool.** It doesn't really matter what processes, methodologies, and practices you apply. What is important is to shape the implementation around agile values and principles with **the goal of gaining Agile mindset and delivering business results or value.**

Values and Principles

Methodology
Framework
Process
Practices
Tools

Though there is value in the items on the right, we value the items on the left more

There is nothing called 'Agile methodology'. Remember Agile is a set of values and principles

Agile Manifesto

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 & interactions	**over**	Processes & 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.

Source: www.agilemanifesto.org

Agile Manifesto Principles

Satisfy the Customer	Welcome Change	Deliver Frequently	Collaborate Daily
Support & Trust Motivated Teams	Promote Face-to-Face Conversations	Deliver Working Software	Promote Sustainable Pace
Promote Technical Excellence	Maximize Through Simplicity	Have Self-Organized Teams	Reflect & Adjust Regularly

Source: www.agilemanifesto.org

My favorites

#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

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

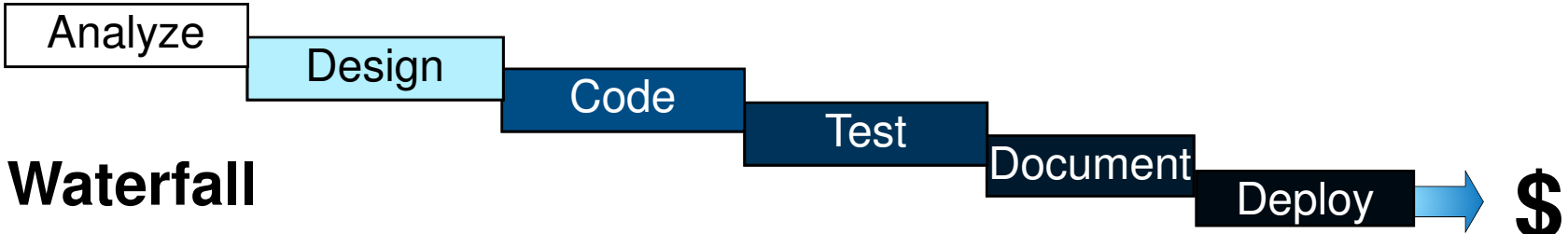
#6 The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**

#7 **Working software is the primary measure of progress**

#10 Simplicity--**the art of maximizing the amount of work not done**--is essential

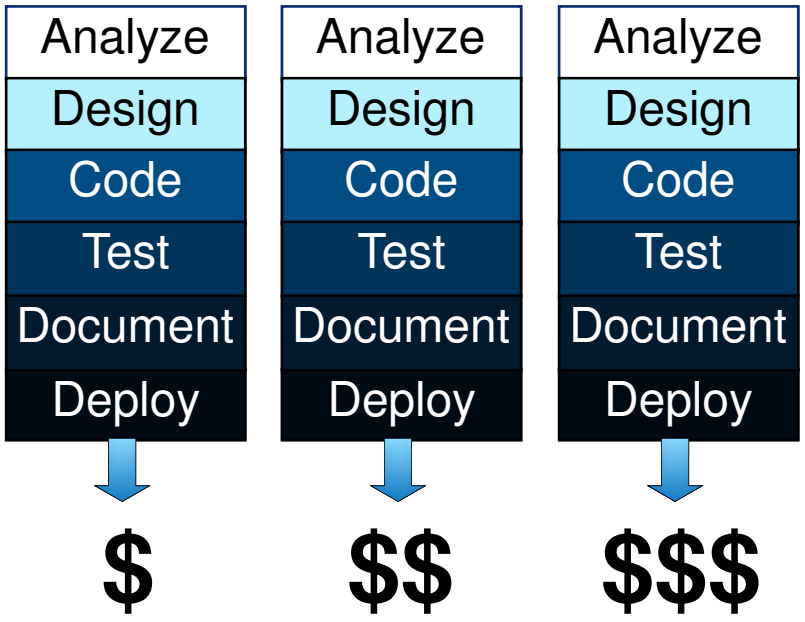
Agile vs Traditional

Delivering Business Value: Agile vs Waterfall



Waterfall

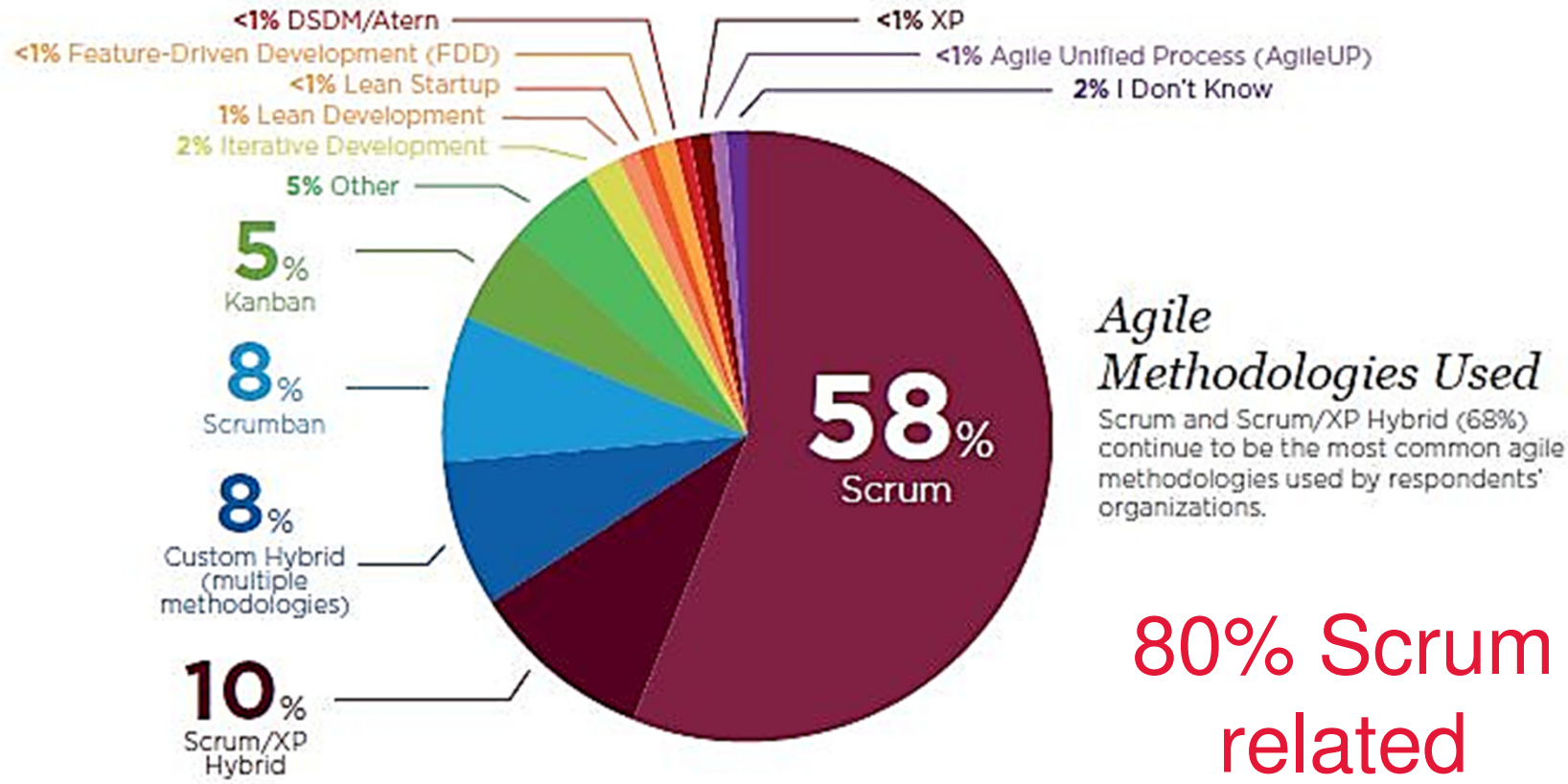
Invest up front, only realize value at end, assuming value proposition hasn't changed



Agile

- **Incrementally** – opt to extend, redirect or cancel at a very granular level
- Deliver & **realize value steadily**
- **Validate designs** with users & customers
- Continuously **adapt to risk and change**
- **Integrate** early & often

Agile Methods and Practices

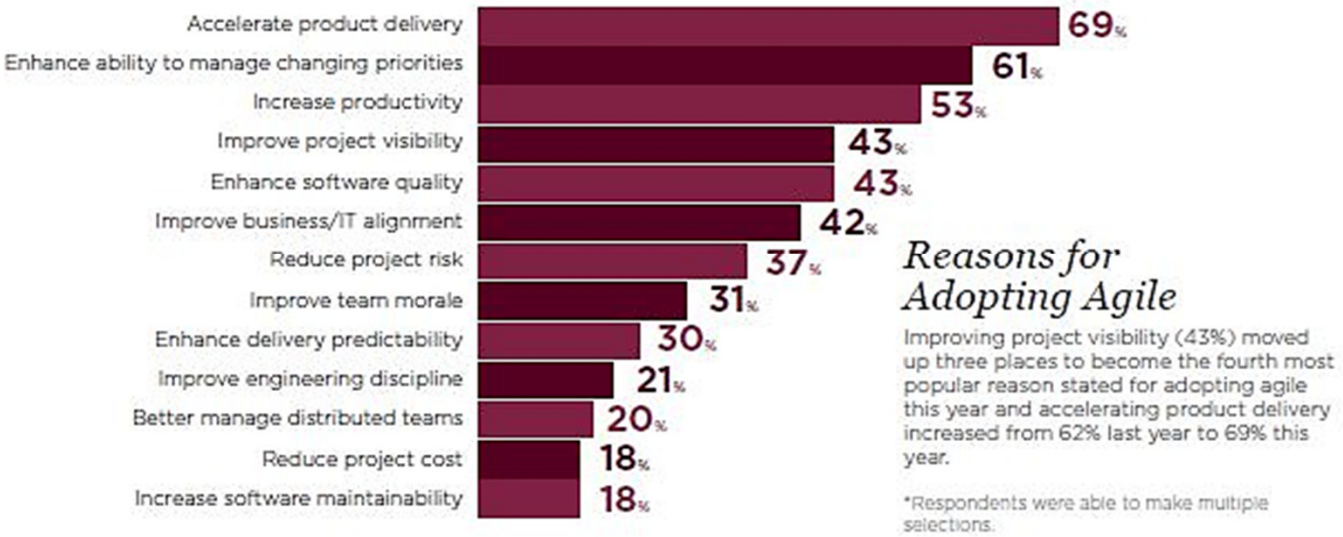


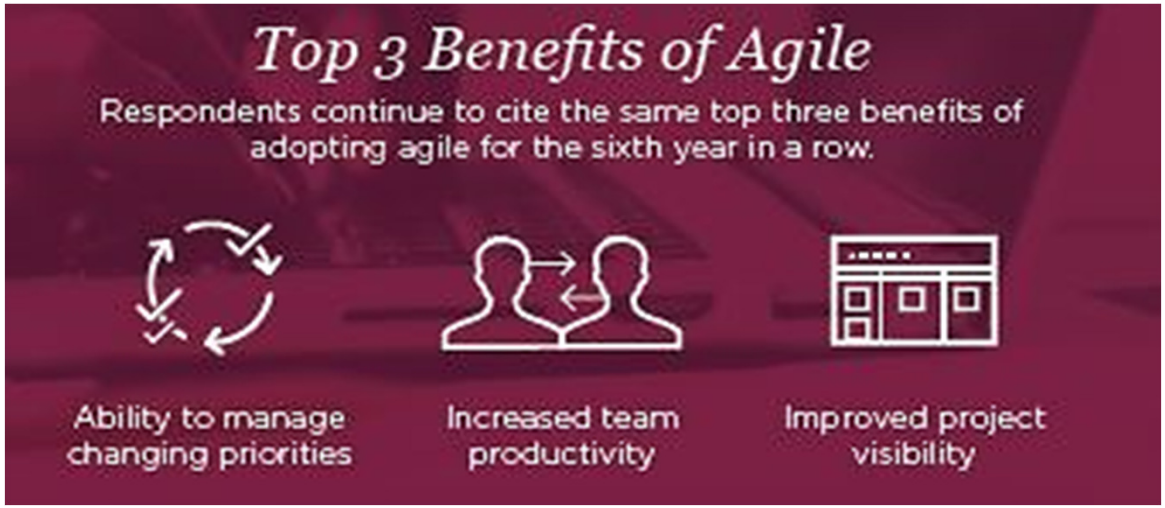
80% Scrum related

Agile development is an umbrella term that describes several agile methodologies to handle IT teams and projects.

Source: 11th State of Agile Report

Reasons for adopting Agile





Source: 11th State of Agile Report

Basic Concepts

Some Basic Terminology

Scrum	Definition
Sprint	Fixed-length period of time (timebox)
Release	Release to production
Sprint/Release Planning	Agile planning meetings
Product Owner	Business representative to project
Retrospective	“Lessons learned”-style meeting
Scrum Master	Process facilitator and helps team perform
Development Team	Empowered Cross-Functional team
Daily Scrum	Brief daily work update meeting

Incremental and Iterative Delivery

Iterating allows you to move from vague idea to realization. Going from **rough to polished**

1 2 3 4 5

Incrementing is more about delivery.

1 2 3 4 5

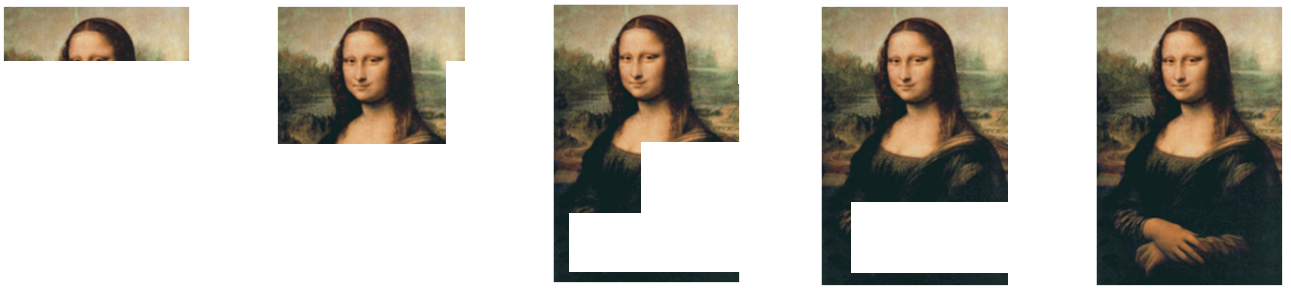


Image Credit: Jeff Patton

?

Scrum Overview

Scrum in a nutshell

- Scrum is an leading agile framework that allows us to focus on delivering the highest business value in the shortest time
- It allows us to rapidly and repeatedly inspect actual working product
- The business sets the priorities. Teams self-organize to determine the best way to deliver the highest priority features
- Anyone can see real working product and decide to release it as is or continue to enhance it for another sprint

Scrum Flow
For One Sprint

Scrum Flow

Product Owner

The Team

Scrum Master

The Team

The Team + Chickens

The Team

*Inputs from Executives, Team, Stakeholders, Customers, and Users (**Project Business Case and Vision**)*

Team selects starting from the top as much as it can deliver by the end of sprint

Planning and Grooming

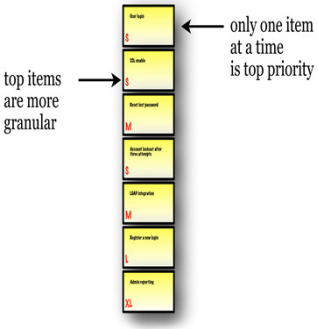
Daily Scrum Meeting
Team commits and work on the committed PBI

Sprint (1-4 weeks)

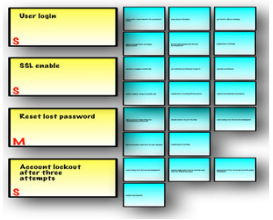
Demo of the product
Review of Work and
Decide on next steps

Review

Retrospective



Product Backlog →



Sprint Backlog

Release Planning Schedule

Potentially Releasable Project

Scrum Roles

The Scrum Team

Product Owner

1. Sets Goals and Product **Vision**
2. Create & Maintain the **Product Backlog**
3. **Approves** product
4. **Prioritize** the Product Backlog
5. **Attends** meetings

Scrum Master

1. **Facilitates** (team decision and process)
2. Helps team **Perform**
3. Detects and strive to **Remove Impediments**
4. Achieve **Transparency** and Visibility
5. Promotes **Engineering Practices**

The Development Team

1. **Cross-functional**
2. **Commits** to sprint
3. **Self-organizing**
4. Ideally in **one room**. Most Successful with **long-term, full-time** membership
5. **3 to 9** members

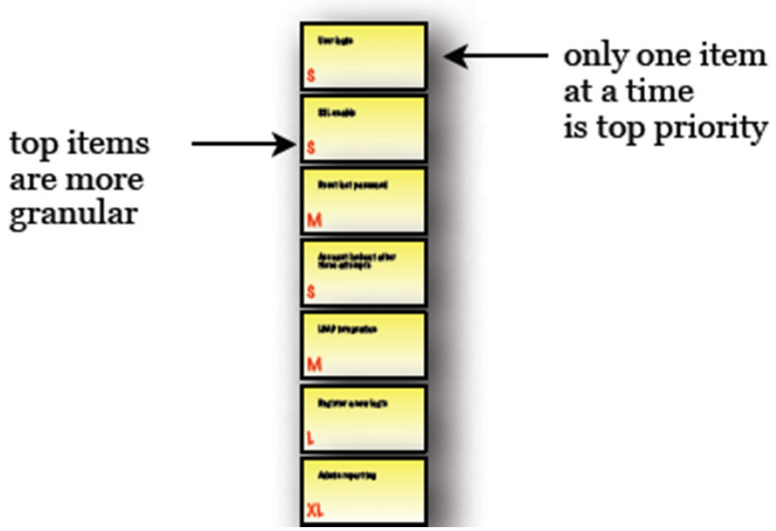
Artifacts

What do you notice about this chart?

Story	Priority	Points
Story A	1	3
Story B	2	8
Story C	3	5
Story D	4	1
Story E	5	3
Story F	6	3
Story G	7	5
Story H	8	13
Story I	9	21
Story J	10	8

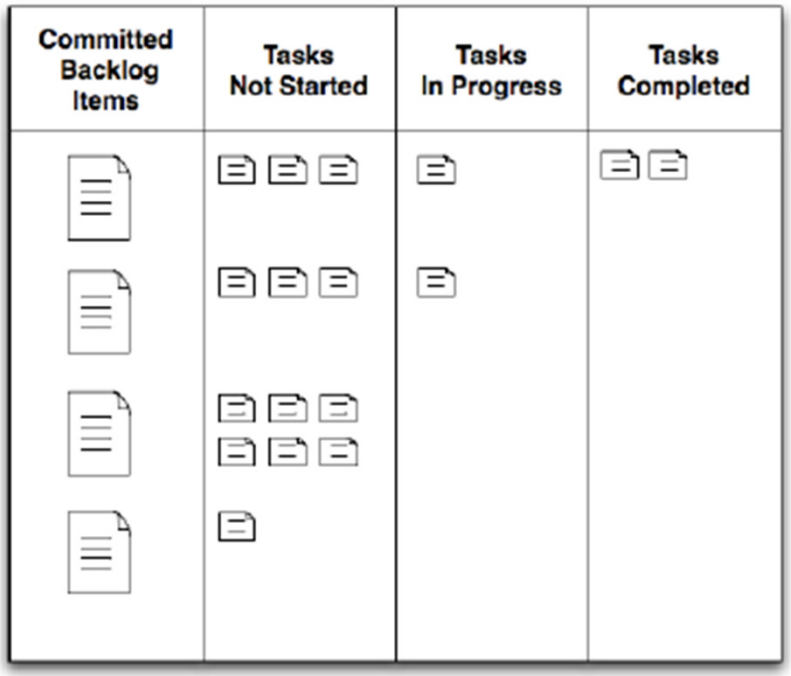
Product Backlog

1. **Force-ranked** list of desired functionality
2. **Visible** to all stakeholders
3. Any stakeholder (including the Team) can add items
4. Constantly **re-prioritized** by the Product Owner
5. **Items at top are more granular** than items at bottom
6. Maintained during the Backlog Refinement Meeting



Sprint Backlog

1. Consists of **committed PBIs** negotiated between the team and the Product Owner during the Sprint Planning Meeting
2. **Scope** commitment **is fixed** during Sprint Execution
3. **Initial tasks** are identified by the team during Sprint Planning Meeting
4. Team will discover additional tasks needed to meet the fixed scope commitment during Sprint execution
5. **Visible to the team**



Sprint Backlog →

#6 Law of Chaos

Law of the edible elephant: *The only way to eat an elephant is one bite at a time.*

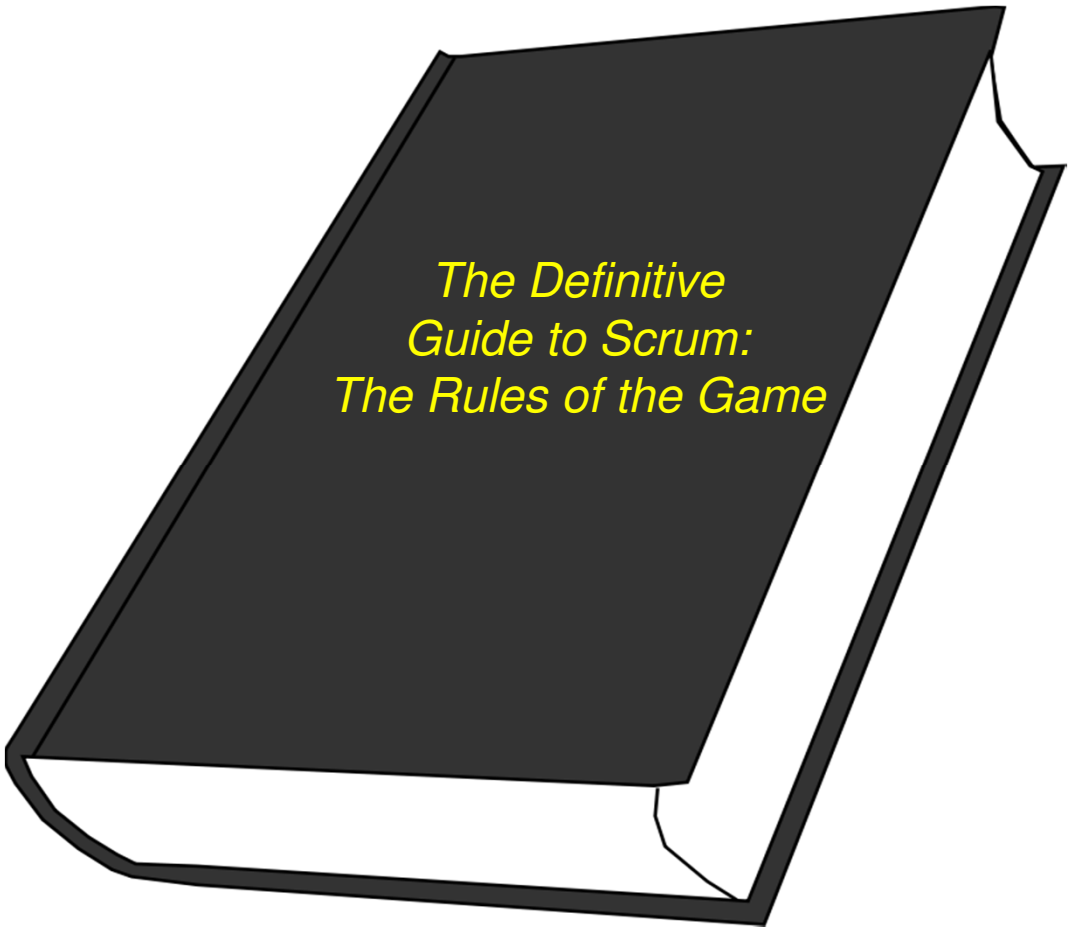
Ceremonies

Scrum Ceremonies

Ceremony	Time Box	Input	Output	Value
Backlog Grooming*	<1 hr	Draft User Stories, Epics from Product Owner	Finalized User Stories Technical Stories Ranking for top PBIs	Product Backlog & Team are ready for Sprint Planning
Sprint Planning	2 - 8 hr	Ranked Product Backlog with Acceptance Criteria	Sprint Backlog: •Selected stories + estimates •Tasks + estimates	Team has a plan to implement Sprint Backlog
Daily Stand-Up	<15 min	In-progress Tasks	Tasks updated Impediments raised	Team members on same page re: Sprint progress and impediments
Sprint Review	< 1 hr	Demo prepared for completed stories	New Stories, based on review by Product Owner Ranking may be revised	Ensure appropriateness of deliverables
Retrospective	1 - 1.5 hr	Sprint performance data, e.g. Burndown chart	Short list of improvements for next Sprint, with owners	Learn from experience, enable continuous improvement

* Not officially a Scrum Ceremony, but important

The Scrum Bible



Developed and sustained by Ken Schwaber and Jeff Sutherland

https://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf

?

Writing ‘Agile’ Requirements

Introductory Exercise

1. **Find a partner**
2. Start telling them about yourself
3. When they hear something you both have in common, they will say **"Me Too!"** and find a new partner

Requirements and Progressive Elaboration

We've learned that

- Detailed requirements are often wrong and incomplete
- The real details emerge over time, as we work

Scrum formalizes this *progressive elaboration*

- Start with brief written requirements
- Develop improved understanding over time

Users don't know what they want until they see a working product or software - Haumprey's Law

IKIWISI

Progressive Elaboration in Scrum

Work	When	Result
Writes one-line summary	Anytime	Story Title
Writes description of the deliverable	Anytime	Story Narrative
Summarizes what to validate	Anytime	Story Acceptance Criteria
Team reviews the PBI before planning	Backlog Grooming	List of questions, issues
Team estimates PBI	Sprint Planning	Estimates
Team identifies and estimates tasks	Sprint Planning	Task breakdown
Team develop acceptances test	Sprint Execution	Test Cases

User Story

A short narrative description of user-facing functionality that one Team can implement in a few days

Title	Any short and meaningful title
Narrative	
Start with summary in one of these standard forms: As a <Role>, I want to <Action> so that <Benefit> A <Role> can <Action> so that <Benefit> <Role><Action> so that <Benefit> If necessary, elaborate on how user interacts with this new deliverable. Include links to images, screen designs, or other external documents of interest. Make sure that the results are testable.	
Acceptance Criteria	
List specific criteria that must be met for deliverable to be accepted	

Example: User Story

Title	View grades online

Narrative

As a **student**, I want to **see my grades on line** so that **I won't have to go to school** to check if I pass.

Acceptance Criteria

1. Runs in popular web browsers like Mozilla, Chrome and EI11
2. I should see grades for the current semester's grades by subject in a columnar form:

Subject	Grades
English	A
Science	A
Math	B+
Music	B-

What makes a good user story?

INVEST	Description
Independent	Make Stories as independent from each other as possible
Negotiable	Brief description. Details emerge in discussion
Valuable	Users and customers perceive value in the deliverables
Estimable	Domain, technical knowledge allow Team to provide estimate
Small	Team can finish one Story in a few days, several in one Sprint
Testable	Validation criteria and techniques are specified clearly

Are users stories equivalent to requirements?

Requirements, More than Just a Story

Context (Project Vision, Business Case, etc)

Requirements = User Story + Conver-sation + Acceptan ce Criteria + Supporting information

Common Understanding of the need

Exercise – Write User Stories

- Create at least **five s**print-sized User Stories based on your project

- Use the **user story template**

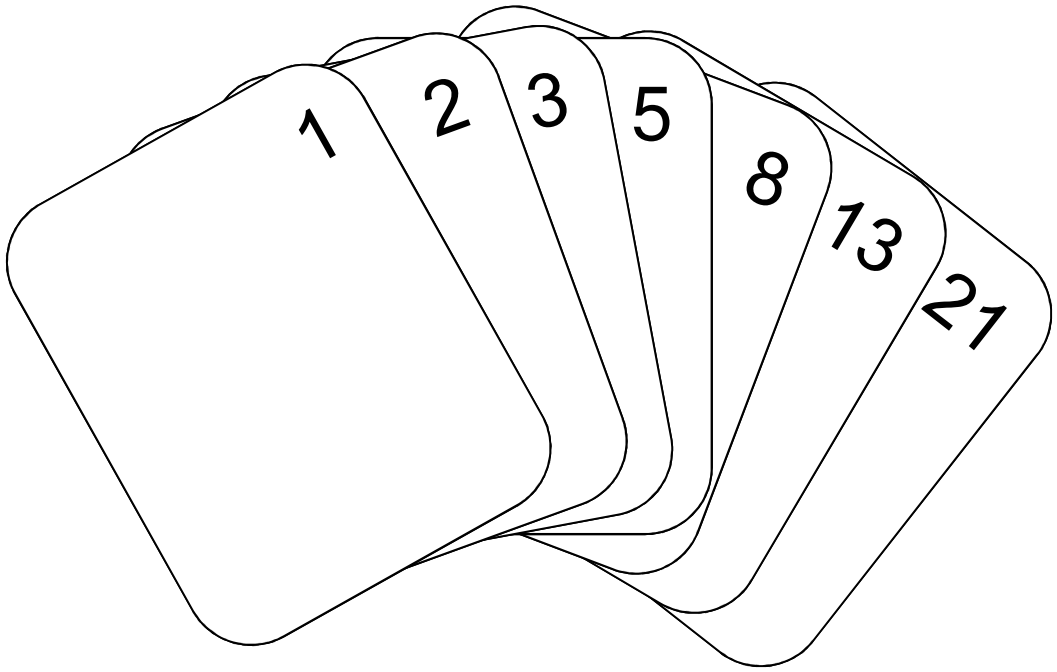
?

Estimation

Estimation

- Each PBI takes time to implement
- We need estimates to plan for sprints
- **Rough estimates will do**
  - –Good enough is good enough
  - –Precision cost time we can't afford and there isn't available anyway

Estimation using Fibonacci



- Fibonacci sequence is a series of number that follow unique integer sequence.
- These numbers generate mathematical patterns that can be found in most aspect of life.
- Patterns that can be found in everything from the human body to the physiology of plants and animals

Planning Poker Instruction

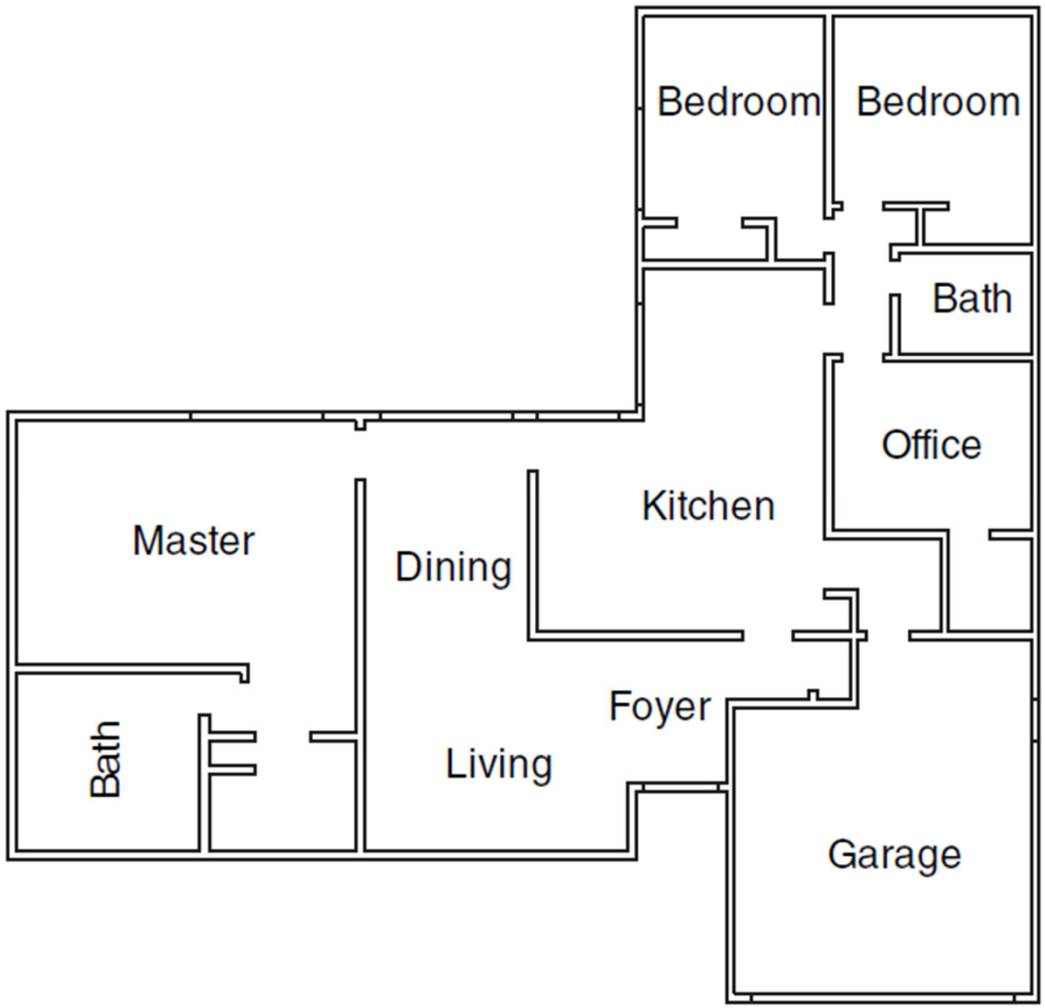
1. Product Owner/Moderator reads PBI to team
2. Team, Product Owner discuss & clarify for up to 10 minutes (time boxed!)
3. Facilitator asks Team members to pick and hide card with estimate
4. Facilitator asks all Team members to show cards
5. If all agree, Facilitator records estimate and moves on
6. Otherwise, Facilitator asks low & high voters to explain the reasoning for their votes
7. Facilitator asks for re-vote, after discussion.
8. Vote up to three times, if necessary to converge

Can't converge?

If close, Team picks "good enough" value

If not close, can't estimate, so ask Product Owner to revise PBI for later estimation

Estimation Practice



How long will it take to paint the house?

Team Exercise: Estimate Stories

Use Planning Poker® decks

1. Story's author will be Product Owner
2. Pick Scrum Master to facilitate estimation session
3. Estimate *effort*
4. Estimate Stories written in previous exercise

- Record final Story estimate on Story's template

Time: 20 minutes

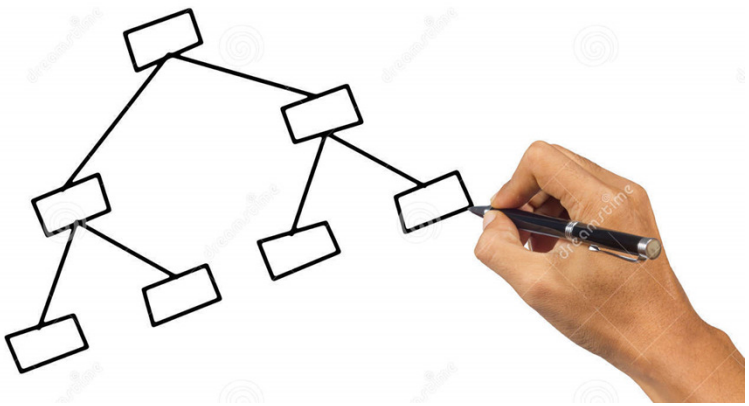
?

Task Breakdown

The Task Breakdown

The Task Breakdown

- Is list of tasks required to implement a PBI
- Is created by the Team at or before start of Sprin



Why we need the Task Breakdown

1. To provide a clear to-do list for the Team
   - Important for Swarming
2. To enable us to track progress effectively
3. To gain additional understanding of requirements
   Progressive elaboration!
4. To provide a more refined estimate of size for Backlog Items

Contents of Task Breakdown

- **Trackable tasks (2—16 hours)**
- ***All* work required to achieve the Definition of Done**
- Design solution, define tests, implement tests, implement solution, deploy to QA, execute tests,…
- **Ordered (roughly) by time**
- We *want* parallel task work, so true ordering is not possible

Exercise: Create Task Breakdowns

1. **Review sample Story and Defect breakdowns**
2. **Use task-breakdown template**
3. **Create a Task Breakdown for the Story you developed earlier**
4. Tasks should include all implementation, testing, and bug fixing work
   Do *not* estimate tasks at this point
5. **Let's review by team**

Time: 10 minutes

Estimating Tasks

- **Planning Poker provided initial PBI estimate**
- **Task-breakdown improves understanding of work**
- **Team estimates all tasks, in units of Person-Hours**
  - "How many hours will one person need to finish this task?"
- **Estimate with informal team discussion**
  - Overhead of Planning Poker not justified
- **Sum of task estimates provides improved PBI estimate**
  - Divide total person-hours by 8 to get Person-Days

Team Exercise: Estimate Tasks

Use Task Breakdowns previously developed

1. Discuss each task, estimate it in person-hours

2. Put estimates on the template, by the tasks

3. Compute new Story estimate

a. Sum task estimates for each Story to provide a task total

b. Divide by 8 to get Person-Days

c. Update Story template with revised Story estimate

Time: 15 minutes

?

Planning for Sprint

The Sprint

- **Smallest unit of scheduled working time for a project**
  - Fixed-length period during which work for a set of PBIs is done
  - Larger cycles are built out of multiple Sprints
- **Time-boxed interval, typically 2—4 weeks, focused on a particular type of work**
  - Boundaries are firm!
  - Sprints are not extended to finish work
- **Sprints should have standard**
  - Experiment, then standardize

Defining Velocity

- Velocity is a crucial concept in Sprint Planning
- We *define* Velocity to be the amount of work done in a Sprint
- For a completed Sprint:

  Velocity = sum of estimates of PBI's completed in the Sprint
- Same PBI estimates used in planning for that Sprint
  - Not defined based on Task estimates
  - Items started but not finished in the Sprint do not contribute to the Sprint's Velocity
- We need a forecast of a future Sprint's Velocity in order to plan that Sprint

The Three Phases of Planning Sprints

1. Develop the Sprint Schedule

- Activities, milestones
- Schedule usually repeated without modification

2. Forecast Sprint Velocity

- Responsibility of Scrum Master

3.Allocate work to the Sprint

- Estimated by Team
- Facilitated by Scrum Master
- Requirements supplied by Product Owner

Introduction to Sprint Scheduling

Schedule of activities includes

- Sprint Planning Meeting*
  - Part 1: Estimate PBIs, create initial Sprint Backlog
  - Part 2: Create Task Breakdowns, finalize Sprint Backlog
- Daily Stand-Up Meeting
- Backlog Grooming (for next Sprint)
  - Team & Product Owner review, refine, fill gaps in Product Backlog
- Sprint Review Meeting
  - Retrospective Meeting Capture lessons learned, plan for improvement

* Product Owner must be present for Part 1, but usually does not attend Part 2

Sample Two-Week Sprint
(Planning for sprint in yellow boxes)

	M	T	W	TH	F	M	T	W	TH	F
8am	Sprint Planning									
9am										
10am										
11am										
12nn										
1pm										
2pm										
3pm										Sprint Review
4pm			Backlog Grooming					Backlog Grooming		Retrospective
5pm										

Any question?

Four Techniques for Forecasting Sprint Velocity

- **Estimate how much work Team can do in the Sprint**
  - Based on Team membership, Sprint duration.
  - Technique depends on choice of units (Relative or Absolute)

- **Should be informed by history**
  - Revise assumptions based on experience

- **For Relative units, methods include**
  1. Est. Velocity = Same as last Sprint
  2. Est. Velocity = Average of last three Sprints

- **For Absolute units, methods include the above and**
  3. Est. Velocity = people × days × focus factor
  4. Estimate from detailed resource model (follows)

Scrum Master estimates prior to Sprint Planning Meeting

Forecasting Velocity via Resource Model

- **Determine available working time in Sprint, per person**
- **Determine effective resources of team**
  - Subtract meeting times, vacation, holiday time per person
  - Estimate availability per person
- **Compute working time**
  - Per person: Availability * (workdays – time off)
  - Per team: Sum of per-person capacities
- **Maximum availability is 75%**
  - Leads have 50% or less
  - Shared people have much smaller numbers per Team

DEFINITIONS:

Availability
Fraction of workday an individual can work on PBIs
•Excludes scheduled Sprint activities

Net Team Resources
Effective number of full-time Team members

Sample Velocity Forecast for One-Week Sprint

Assume 8 hours in Scrum Meetings per Sprint (of 40 hours)

Team Member	% Avail	Hrs Off	Hours
Nathan	25%		8
Lisa	40%		12.8
Audrey	75%		24
Ivan	75%		24
Walline	75%		24
Yam	50%	16	8
Joshua	75%		24
Velocity	15.6	Team Hours	124.8

1. Hours = Avail. * (Net work hours in Sprint – Hours off)
2. Team Resources = Team Hours / (Work hours in Sprint)
3. Velocity = Team Hours / 8

Limitations, Benefits of Velocity Forecasting

Limitations

- **These forecasting methods assume work is generic**
- They do not account for specialization of resources
- They will not work if specialization is extreme

- **Analysis is optimistic**
- Best-case scenario, because it omits specialization

Benefits

- **Straightforward and easy to understand**
- •Simplicity has value! **Bounds what can be done**
- •Work that exceeds the optimistic estimate is unlikely to be completed

Velocity forecasting is an effective planning technique for a Team of Generalizing Specialists

Exercise: Forecast Velocity

1. Use template
2. Estimate Velocity for each of 2 teams of 4 people, with two-week Sprint
3. Let's review

Time: 25 minutes

?

Tracking Progress

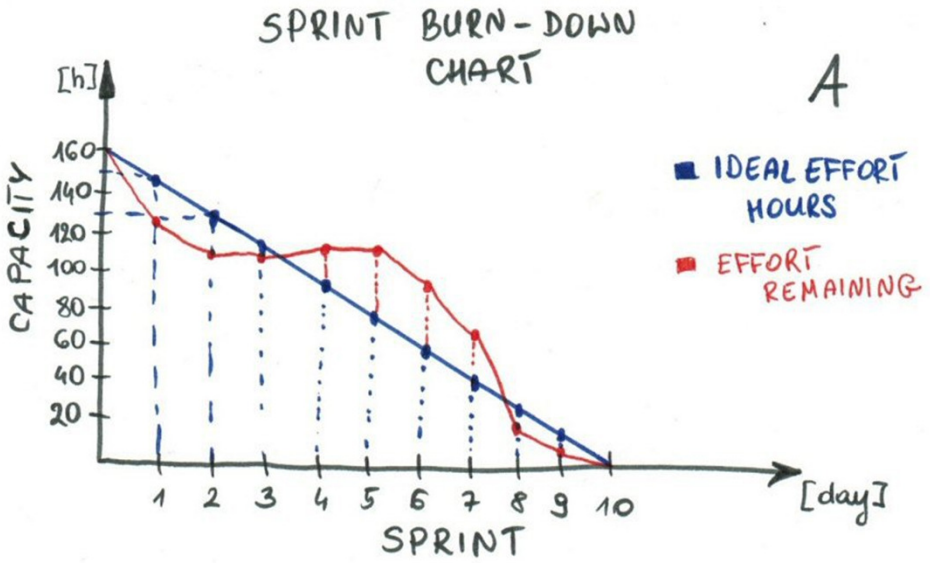
Acquiring Task Information

- **What we want to know about Tasks**
  - Estimate for task effort (person-hours), from Task Breakdown
  - Status: Not Started, In Progress, or Complete
  - If In Progress, the Effort Remaining (maybe)
    - Often not reliable. Beware of “1 hour remaining” for 3 days!
  - After completion, the actual time required (maybe)
    - Required for billable hours, hard to get from Team otherwise
    - Can be used to refine estimation process over time
- **•When we need to know these things**
  - Now (whenever status changes)
  - Each Team member is responsible for providing status information!
  - Scrum Master has to prompt occasionally…

Summarizing Progress via Burndown Chart

Red: Amount of remaining work (“Estimate to Complete”)
Sum of estimates for uncompleted tasks

Blue: Plan
Goes to zero at end of Sprint
Usage: Primarily for time span of a Sprint



Exercise: Make a Burndown Chart

1. **Use Burndown chart template**
2. **Draw ideal progress line from 90 at Day 0 to 0 at Day 10**
3. **Use these "Work Remaining" figures for column chart**

Day	Work Remaining
0	90
1	85
2	80
3	74

Day	Work Remaining
4	68
5	60
6	52
7	44

Time: 5 minutes

Question: Is planned work likely to complete in Sprint?

•What do you predict for the deviation from plan?

?

Time box meetings

Backlog Grooming

Purpose

- Ensure Product Backlog & Team are ready for future Sprint Planning Meetings
  - **When: 1—2 hours per week**
  - **Who: Team, Product Owner**
- **Actions in meeting**
  - Provide feedback on clarity, quality, acceptance criteria, dependencies, ranking of Stories
  - Identify “holes” (esp. technical) for which Stories must be written
  - Break Epics into Stories
  - Do long-term technical (architecture / infrastructure) planning
  - Identify, update list of, risks / threats / issues, & incorporate into Backlog

Follow-up actions

- Team, Product owner write or revise Stories, as needed

Sprint Planning Meeting

Purpose: Assign PBIs to Sprint Backlog

- **Scrum Master facilitates, enforces selected time box**
- E.g., 1 hour, if Team has reviewed PBIs carefully in advance

Part 1:

For each Product Backlog Item (PBI), in rank order

1. Product Owner reads PBI to Team
2. Team discusses, asks Product Owner to clarify details
3. Scrum Master facilitates estimation & records results
4. Scrum Master adds PBI to Sprint Backlog
5. Planning is finished when Sprint Backlog is filled to Velocity limit

Part 2:

- Team creates Task Breakdowns for Sprint Backlog items
- Revise scope of Sprint Backlog as needed based on Task estimates

The Sprint (Each Day's Major Activities)

Purpose: Implement PBIs in Sprint Backlog

- **Scrum Master monitors work, facilitates issue resolution**
- **Team members swarm to implement PBIs in rank order**
  - Ask Product Owner to clarify requirements
  - Ask Scrum Master to resolve issues the Team cannot resolve
- **Team members update status of each task**
  - On starting, finishing, revising "to-do" effort, …
- **Team members don't start PBIs they can't finish in Sprint**
  - Maintain discipline of finishing what is started!

Daily Stand-Up Meeting

Purpose: Promote common understanding of Sprint status, and identify issues to be resolved

- **Scrum Master facilitates, enforces 15-minute time box**
- **Team members, Scrum Master, Product Owner attend**

Agenda

1. Scrum Master shows burndown chart, describes progress
2. Each Team member describes
   - What I've done since the last Daily Stand-Up meeting
   - What I plan to do before the next Daily Stand-Up meeting
   - What issues I'm facing that I need help to resolve

In meeting

- Decide who will collaborate to resolve each issue after the meeting ("sidebar discussions")

Sprint Review Meeting

Purpose: Confirm acceptability of implementations

- **Scrum Master facilitates, enforces selected time box**

Agenda

- **1.Team demonstrates finished PBIs to the Product Owner**
  - Team members decide who will do the demonstrations.
  - One person does all; round-robin style; etc.

2.Product Owner provides final decision on whether implementations are acceptable for release

  - If not, then they are not released
  - Should be rare, since PO monitors & evaluates throughout Sprint.

After

- Product Owner writes Stories for changes to implementations

Retrospective Meeting

Purpose: Learn from experience, and improve

- An example of “kaizen” concept of continuous improvement
- **Scrum Master facilitates, records, enforces time box**
- Say, 60 minutes total: 30 for recording, 30 for discussion

Agenda

1. Review status of work items from previous Retrospective
2. Team members, Product Owner, Scrum Master answer
   - What went well, that we should do again?
   - What would we like to be better?
3. Specify follow-up actions
   1. Prioritize improvements
   2. Select top few to address
   3. Select owners to drive improvements

How to Conduct a Retrospective Meeting

How to capture "went well, want to be better" items

- *Don't* ask to give info one at a time
  - Sequential query is slow
  - Risks anchoring
- *Do* ask everyone to write items on sticky notes, post on board
  - Parallel data collection is quick
  - Minimizes anchoring

How to define follow-up actions

- Consolidate similar items
- Use multi-voting to rank suggested improvements
- Discuss, re-vote as needed to get consensus
- Ask for volunteers to own each item that requires follow-up effort
- Stop when consensus is that we're tackling enough

Wrap Up