



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SWE 2029 – Agile Development Process

Module - 5

SCRUM

Dr. Rajesh M

*School of Computer Science and Engineering
Vellore Institute of Technology [VIT]
Chennai, India.*

Module	Topics	L Hrs
V	<p>AGILE METHODOLOGIES:</p> <p>Scrum Foundations - Scrum Roles - Scrum Master - Product Owner – Team - Scrum Meetings - Scrum Artifacts - Product Backlog - Sprint Backlog - Burn-down Charts - Scaling Scrum – Manager in Scrum and Product Backlog.</p>	7

Scrum has been used by Popular Industries:

- Microsoft
- Yahoo
- Google
- J P Morgan
- IBM
- SAP
- Cisco
- Nike
- Dell
- Philips
- Siemens
- Nokia
- Capital One
- BBC
- High Moon Studios
- Intuit
- Nielsen Media
- American Express
- BMC Software
- Ipswitch
- John Deere
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting
- Oce

Scrum has been used for:

- Commercial software
- In-house development
- Contract development
- Fixed-price projects
- Financial applications
- ISO 9001-certified applications
- Embedded systems
- 24x7 systems with 99.999% uptime requirements
- the Joint Strike Fighter
- Video game development
- FDA-approved, life-critical systems
- Satellite-control software
- Websites
- Handheld software
- Mobile phones
- Network switching applications
- ISV applications
- Some of the largest applications in use

Characteristics

- Self-organizing teams
- Product progresses in a series of month-long “sprints”
- Requirements are captured as items in a list of “product backlog”
- No specific engineering practices prescribed
- Uses generative rules to create an agile environment for delivering projects
- One of the “agile processes”

The Agile Manifesto—a statement of values

Individuals and interactions

over

Process and tools

Working software

over

Comprehensive documentation

Customer collaboration

over

Contract negotiation

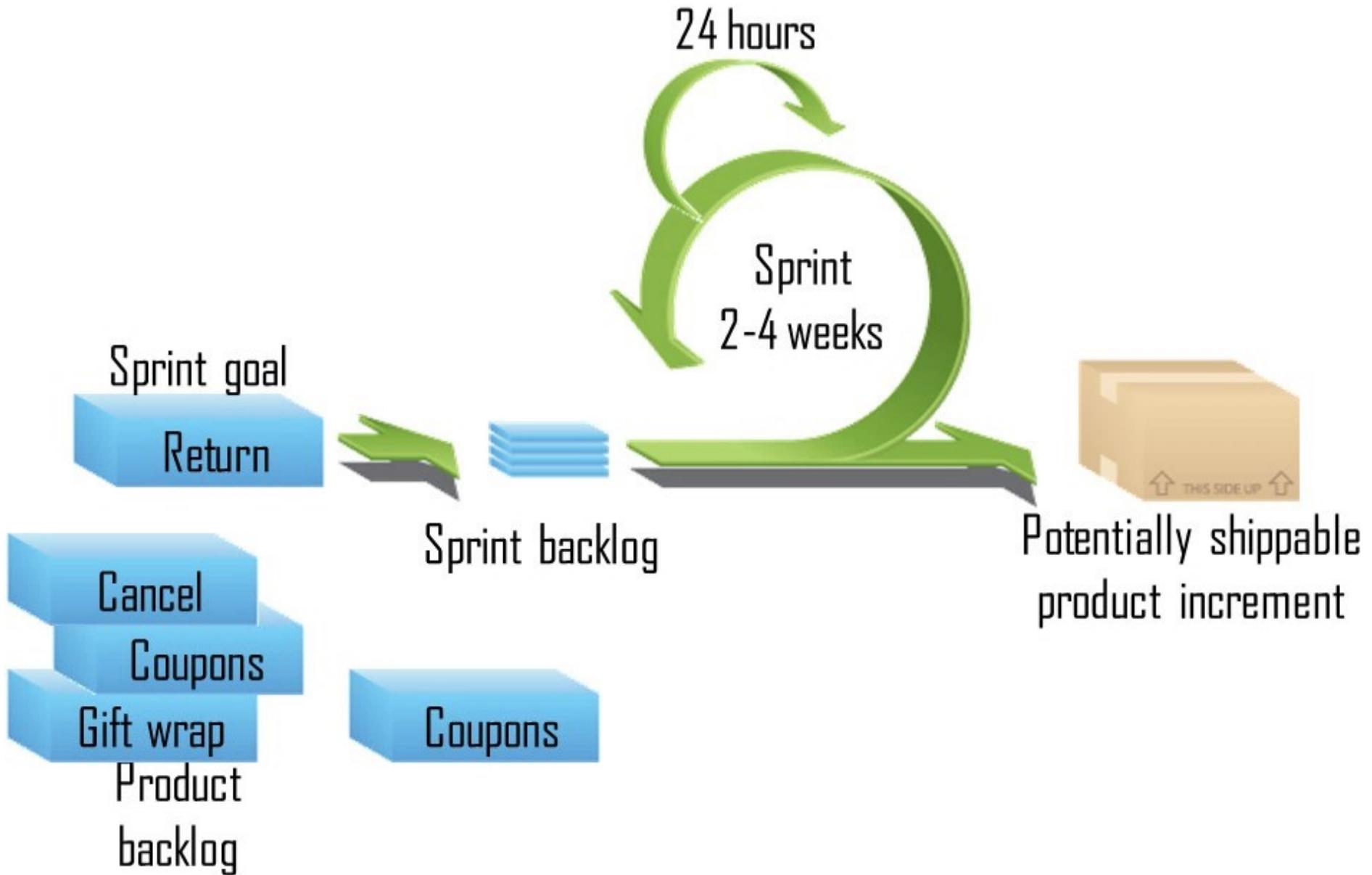
Responding to change

over

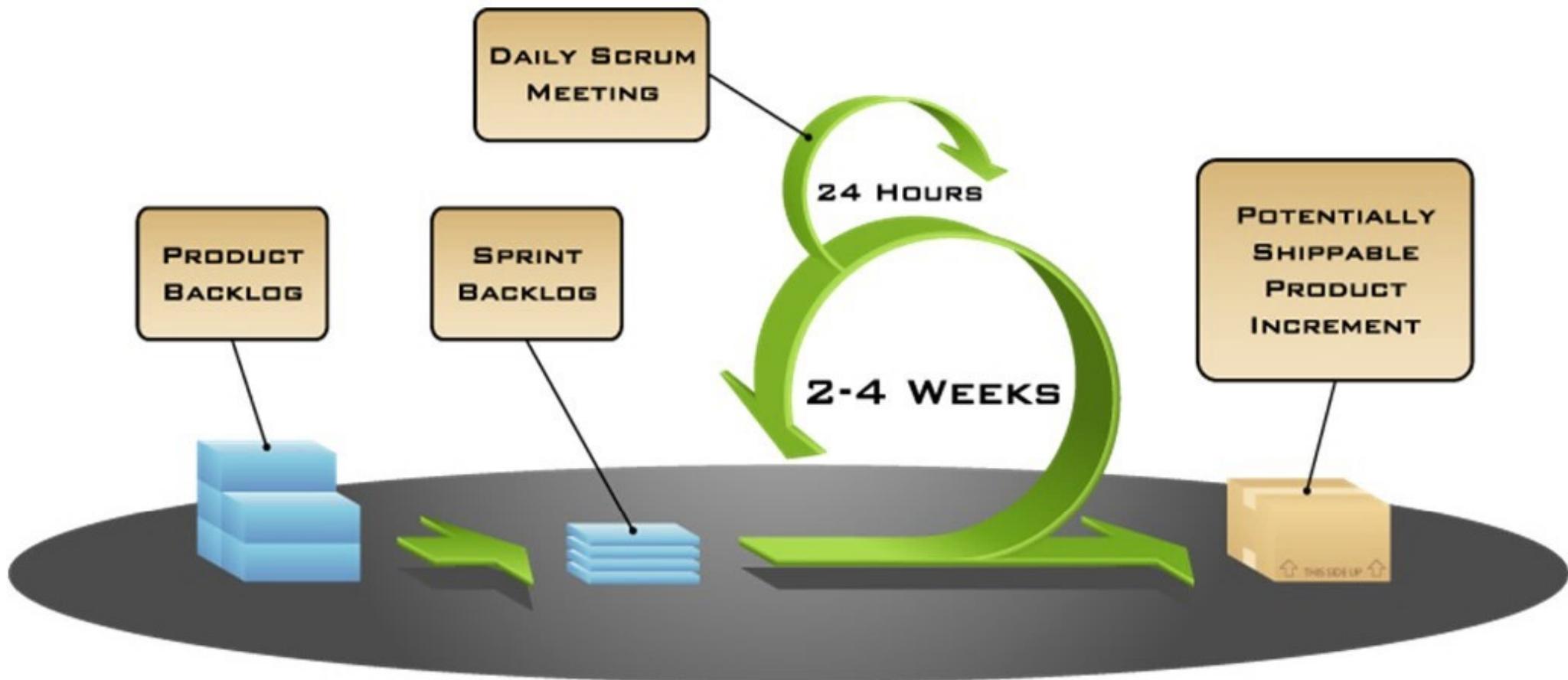
Following a plan

Source: www.agilemanifesto.org

Scrum



Putting it all together



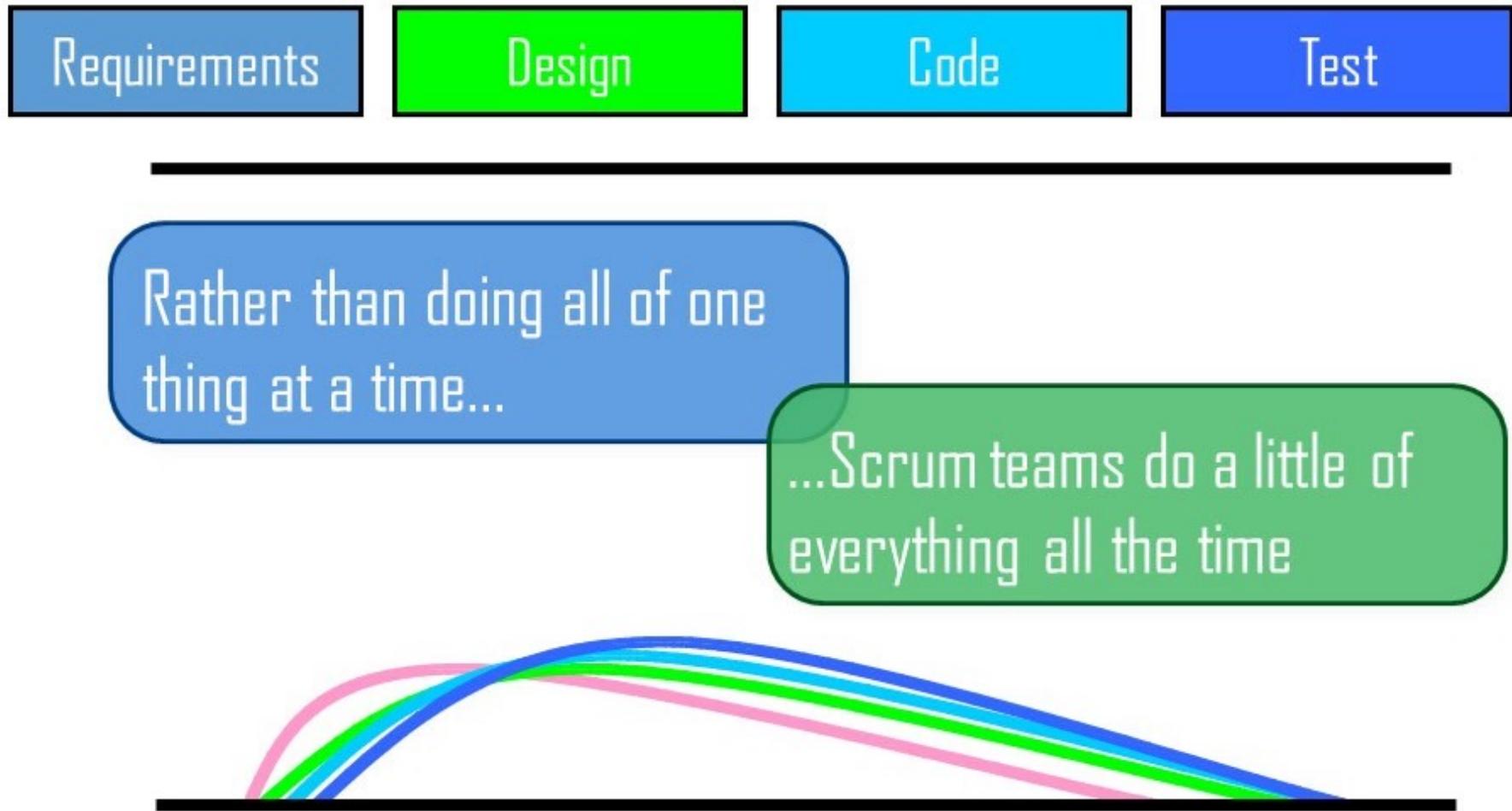
COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Image available at www.mountaingoatsoftware.com/scrum

Sprints

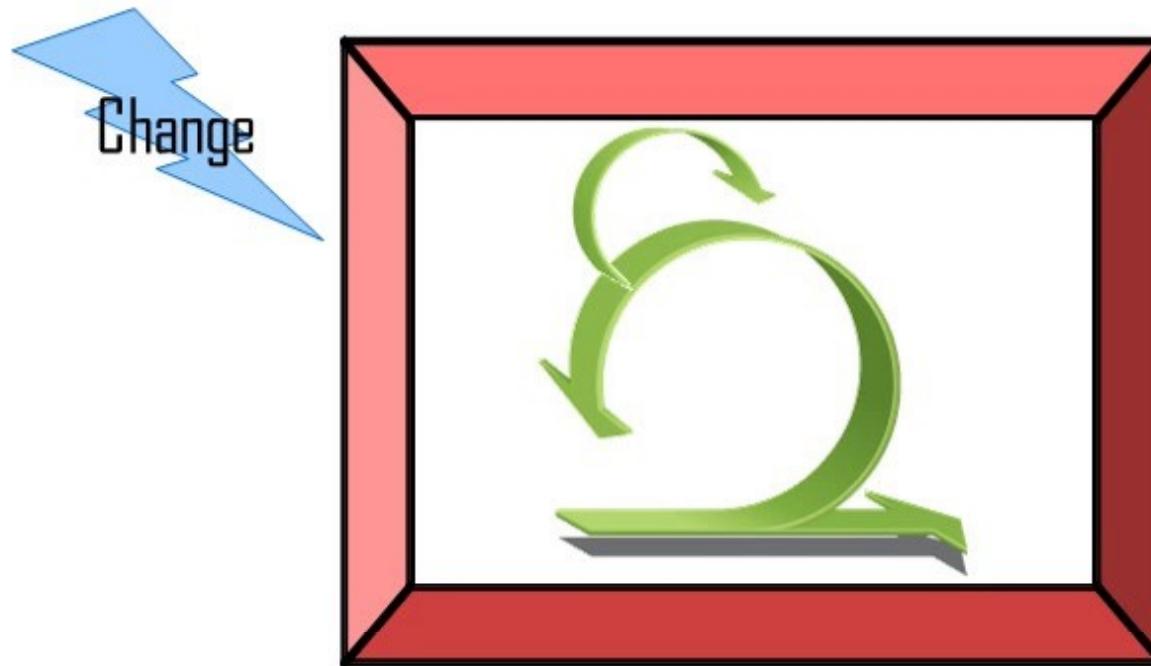
- Scrum projects make progress in a series of “sprints”
 - Analogous to Extreme Programming iterations
- Typical duration is 2–4 weeks or a calendar month at most
- A constant duration leads to a better rhythm
- Product is designed, coded, and tested during the sprint

Sequential vs. overlapping development



Source: “The New New Product Development Game” by Takeuchi and Nonaka.
Harvard Business Review, January 1986.

No changes during a sprint



- Plan sprint durations around how long you can commit to keeping change out of the sprint

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Scrum Roles

Roles

- Product owner
- ScrumMaster
- Team

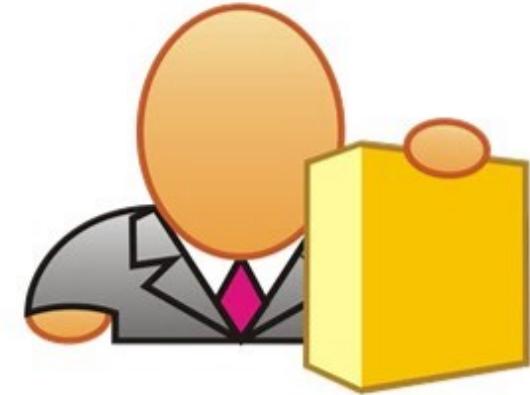
Ceremonies

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

Artifacts

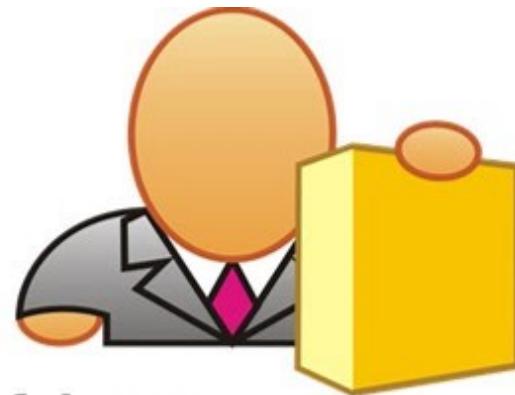
- Product backlog
- Sprint backlog
- Burndown charts

Product owner



- Define the features of the product
- Decide on release date and content
- Be responsible for the profitability of the product (ROI)
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed
- Accept or reject work results

Product Owner



- The Product Owner represents **stakeholders** and is the **voice of the customer**.
- Product Owner is **accountable** for ensuring that the team **delivers value** to the business.
- **Product Owner**
 - **writes** customer-centric items (typically **user stories**),
 - **prioritizes** them, and
 - **adds** them to the **product backlog**.

Note:

- Scrum teams should have **one Product Owner**.
- May also be a member of the development team
- Not recommend this person be Scrum Master.

The Scrum Master



- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences

Scrum Master



- Scrum is facilitated by a Scrum Master –
- Accountable for **removing impediments** for team to deliver sprint goal / deliverables.
- **Scrum Master is not the team leader**, but acts as a **buffer** between the team and any distracting influences.

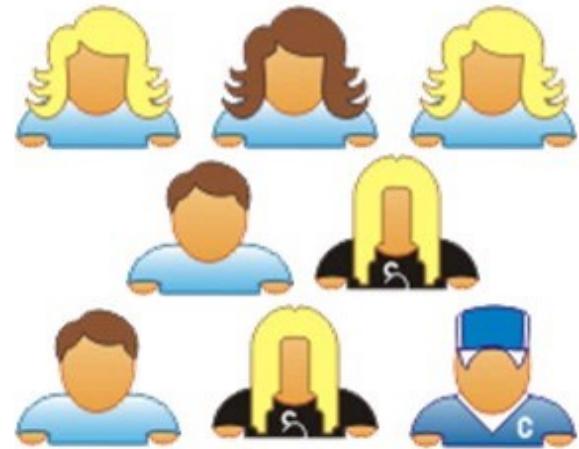
- Scrum Master ensures **process** is used as intended.
- Scrum Master is the **enforcer of rules**.
- Scrum Master's role: **protect** the Team and keep it **focused** on the tasks at hand.

The team



- Typically 5-9 people
- Cross-functional:
 - Programmers, testers, user experience designers, etc.
- Members should be full-time
 - May be exceptions (e.g., database administrator)

The team



- Teams are self-organizing
 - Ideally, no titles but rarely a possibility
- Membership should change only between sprints

Development Team



- The Development Team is responsible for **delivering potentially shippable product increments** at end of each Sprint.
- Team = 3-9 people with cross-functional skills.
- Team does actual work
 - (analyze, design, develop, test, technical communication, document, etc.).
- Team is **self-organizing**, even though they may interface with project management organizations (PMOs).

Scrum framework

Roles

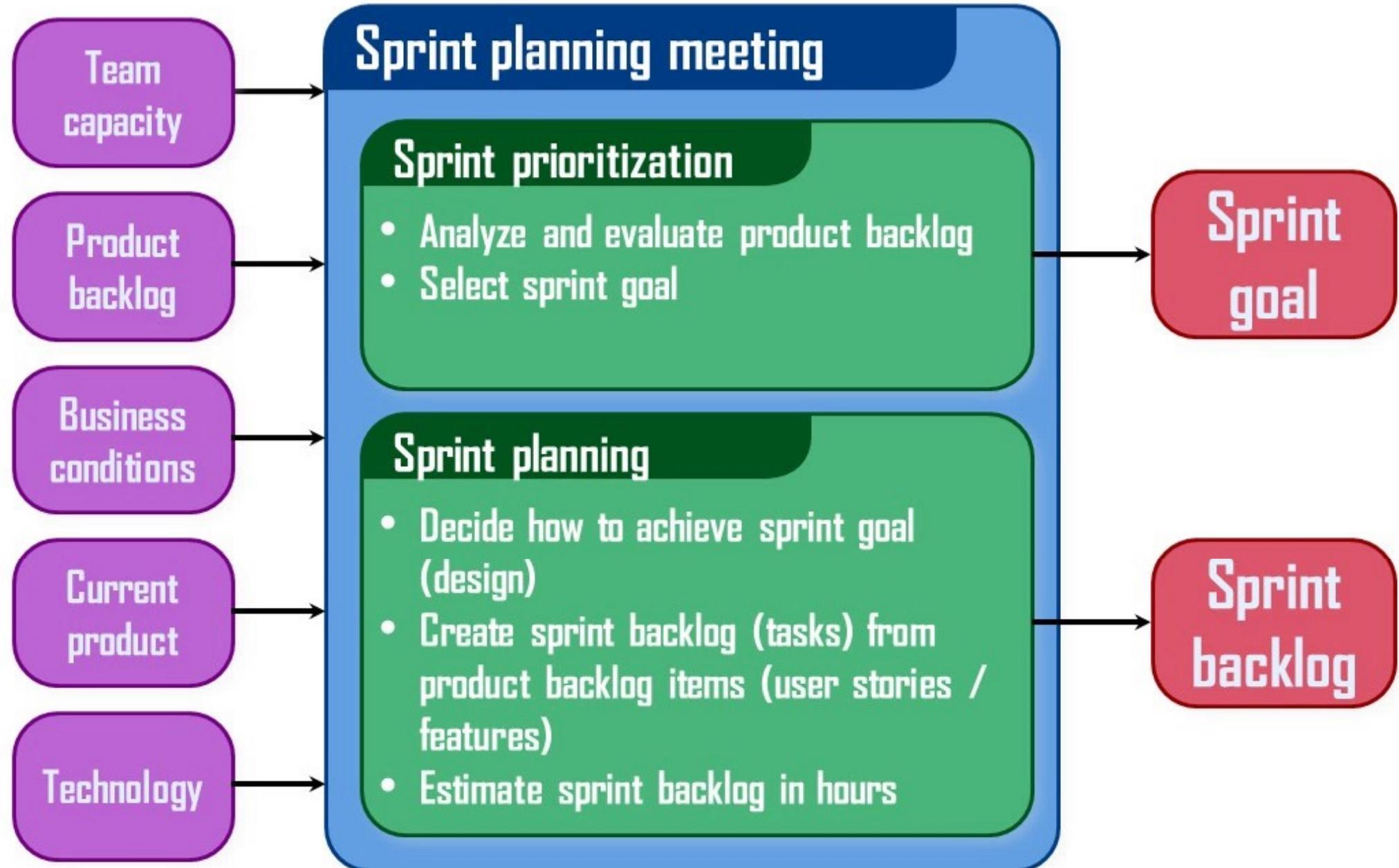
- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts



Sprint planning

- Team selects items from the product backlog they can commit to completing
- Sprint backlog is created
 - **Tasks are identified** and each is estimated (1-16 hours)
 - Collaboratively, **not done alone by the ScrumMaster**
- High-level design is considered

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)

The daily scrum

- Parameters
 - Daily
 - 15-minutes
 - Stand-up
- Not for problem solving
 - Whole world is invited
 - **Only team members, ScrumMaster, product owner, can talk**
- Helps avoid other unnecessary meetings



Everyone answers 3 questions

What did you do yesterday?

1

What will you do today?

2

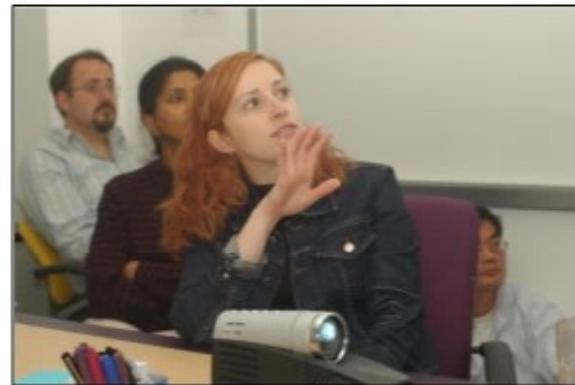
Is anything in your way?

3

- These are *not* status for the ScrumMaster
 - They are commitments in front of peers

The sprint review

- **Team presents what it accomplished** during the sprint
- Typically takes the form of a demo of new features or underlying architecture
- **Informal**
 - 2-hour prep time rule
 - **No slides**
- Whole team participates
- Invite the world



Sprint retrospective

- Periodically take a **look at what is and is not working**
- Typically **15-30 minutes**
- Done after every sprint
- Whole team participates
 - **ScrumMaster**
 - **Product owner**
 - **Team**
 - **Possibly customers and others**

Start / Stop / Continue

- Whole team gathers and discusses what they'd like to:

Start doing

Stop doing

This is just one of
many ways to do a
sprint
retrospective.

Continue doing

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Product backlog

- The requirements
- A list of all desired work on the project
- Ideally expressed such that each item has value to the users or customers of the product
- Prioritized by the product owner
- Reprioritized at the start of each sprint



This is the product backlog

Product Backlog

- **Product backlog** is an ordered list of "requirements" that is maintained for a product
- Contains Product Backlog Items **ordered** by the Product Owner based on
 - considerations like risk,
 - business value,
 - dependencies,
 - date needed, etc.
- **Features** added to backlog commonly written in story format
- The product backlog is the "**What**" that will be built, sorted in the relative order it should be built in.
 - Is open and editable by **anyone**,
 - **Product Owner is ultimately responsible** for **ordering the stories** on the backlog for the Development Team.

Product Backlog

- The product backlog contains rough estimates of both business value and development effort, these values are often stated in **story points** using a rounded Fibonacci sequence.
- Those estimates help the Product Owner to gauge the timeline and may influence ordering of backlog items.
- Example, if the “add spellcheck” and “add table support” features have the same business value, the one with the smallest development effort will probably have higher priority, because the Return on Investment is higher.

Product Backlog

- **Product Owner:** responsible for **the product backlog** and the **business value** of each item listed.
- **Development Team:** responsible for the **estimated effort** to complete each backlog item.
- Team contributes by estimating Items and User-Stories, either in "**Story-points**" or in "**estimated hours**."

A sample product backlog

Backlog item	Estimate
Allow a guest to make a reservation	3
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

Sprint Backlog

Sprint Backlog

- **Sprint Backlog:** list of work the Development Team must address during the next sprint.
- List derived by selecting stories/features from the top of the product backlog until the Development Team feels it has enough work to fill the sprint.
- **Thinking:** This is done by the Development Team asking "Can we also do this?" and adding stories/features to the sprint backlog.
- **History:** Development Team should note **velocity** of previous Sprints (total story points completed from each of the last sprints stories) when selecting stories/features for the **new sprint**.
- Use **number as guide** for "effort" they can complete.

Sprint Backlog

- **Stories/features:** broken down into **tasks** by Development Team
- Should normally be between **four and sixteen hours** of work.
- With this level of detail the Development Team understands exactly what to do, and potentially, anyone can pick a task from the list.
- **Tasks** on sprint backlog are **never assigned**; tasks are **signed up for** by team members during **daily scrum**, according to **priority** and **member skills**.
- Promotes **self-organization** of Team, and **developer buy-in**.
- **Sprint backlog is property of Team**, and all included **estimates** are provided by the Development Team.

The sprint goal

- A short statement of what the work will be focused on during the sprint

Database Application

Make the application run on SQL Server in addition to Oracle.

Life Sciences

Support features necessary for population genetics studies.

Financial services

Support more technical indicators than company ABC with real-time, streaming data.

Managing the sprint backlog

- Individuals sign up for work of their own choosing
 - Work is never assigned
- Estimated work remaining is updated daily

Managing the sprint backlog

- Any team member can add, delete or change the sprint backlog
- Work for the sprint emerges
- 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

A sprint backlog

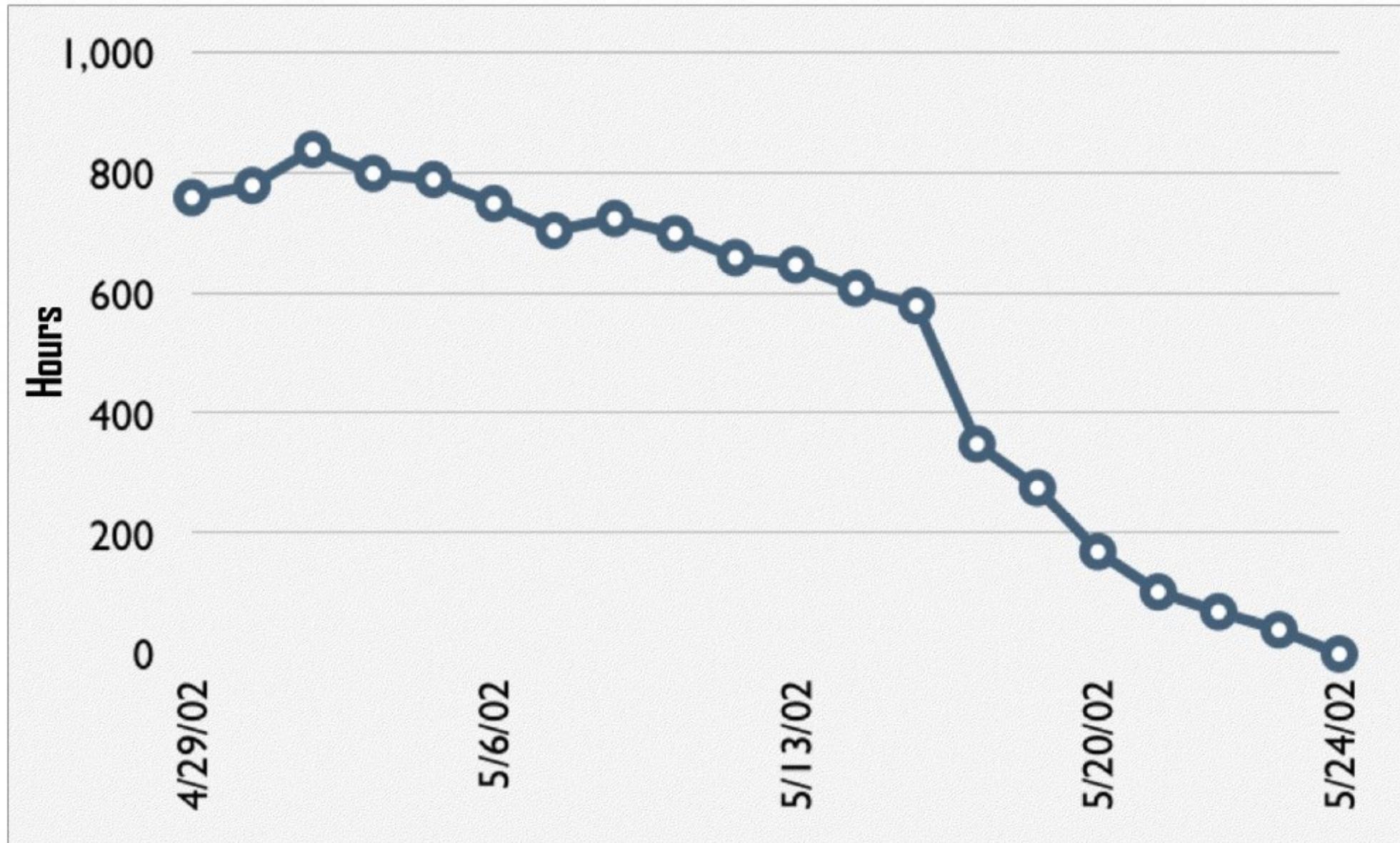
Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the foo class	8	8	8	8	8
Add error logging			8	4	

Burndown Chart

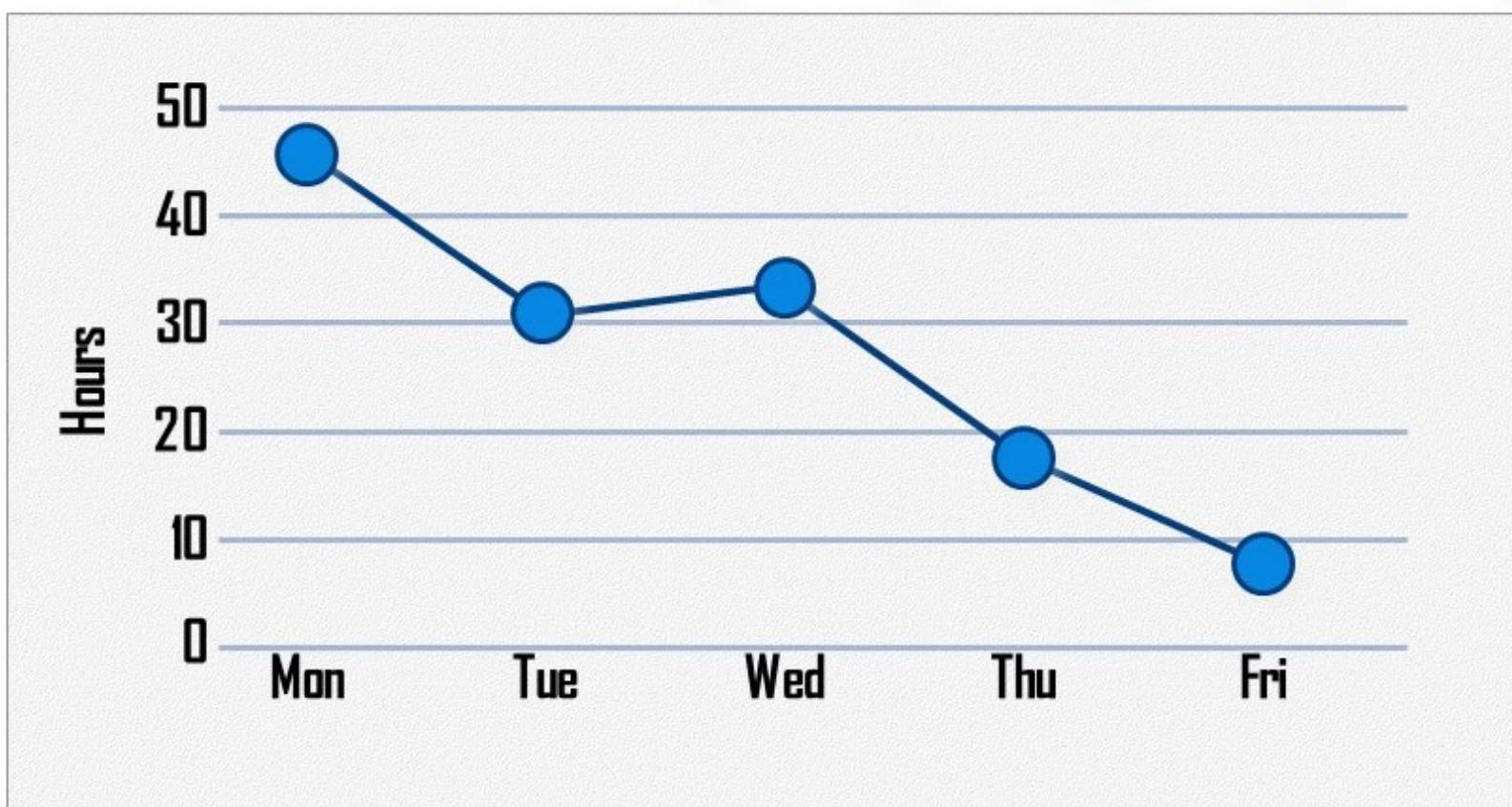
Burn Down

- The sprint **burn down chart** is a publicly displayed chart showing **remaining work** in the sprint backlog.
- Updated every day; gives a simple view of the sprint progress.
- Other types of burn down:
- **Release burn down chart:** shows amount of work left to **complete** the target commitment for a Product Release
 - This normally spans multiple iterations
- **Alternative Release burn down chart:** basically does the same, but clearly shows scope changes to Release Content, by resetting the baseline.
 - This should not be confused with an earned value chart.

A sprint burndown chart

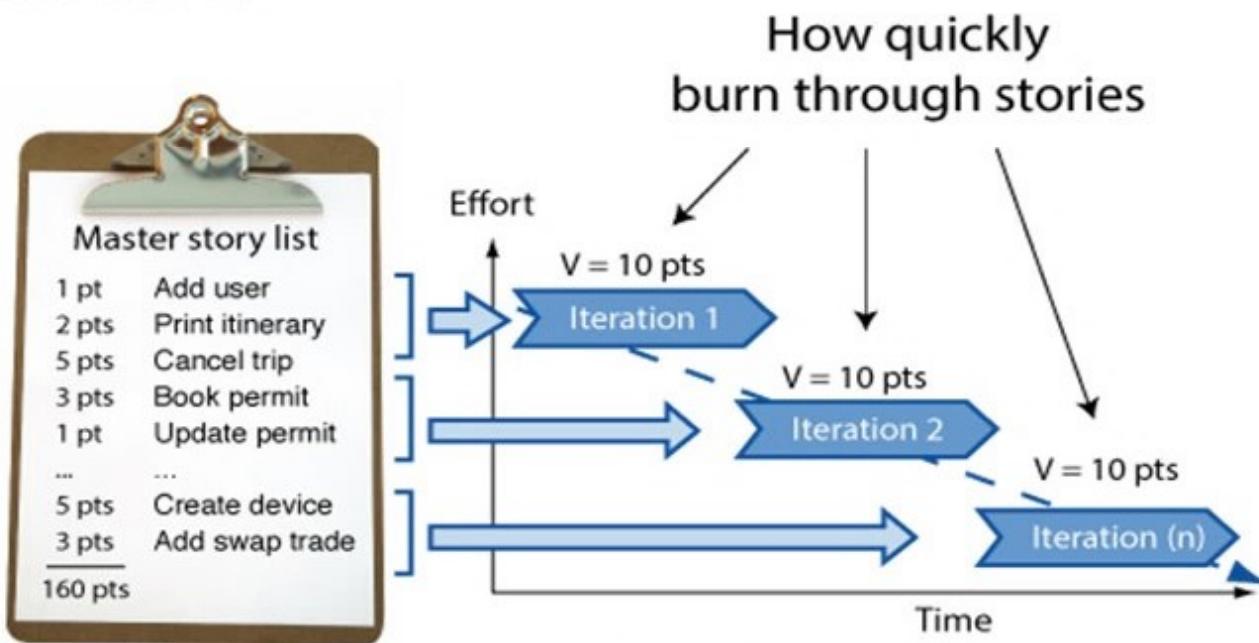


Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	7	
Test the middle tier	8	16	16	11	8
Write online help	12				



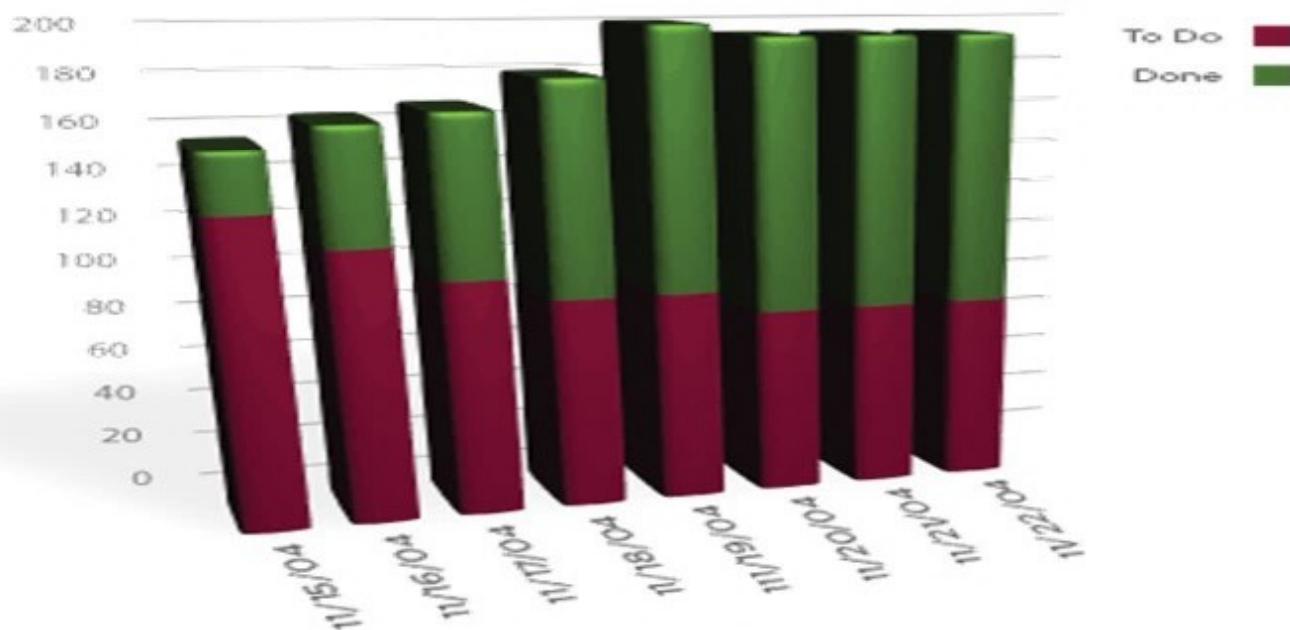
SCRUM - Burndown Charts

- Burn-downs charts are among the most common sprint tracking mechanisms used by Agile practitioners.
- Though their application and usage varies (some plot a burn-down chart using story points, whereas others use task count), plotting burn-down using effort remaining is the most effective and efficient way of using burn-down charts.

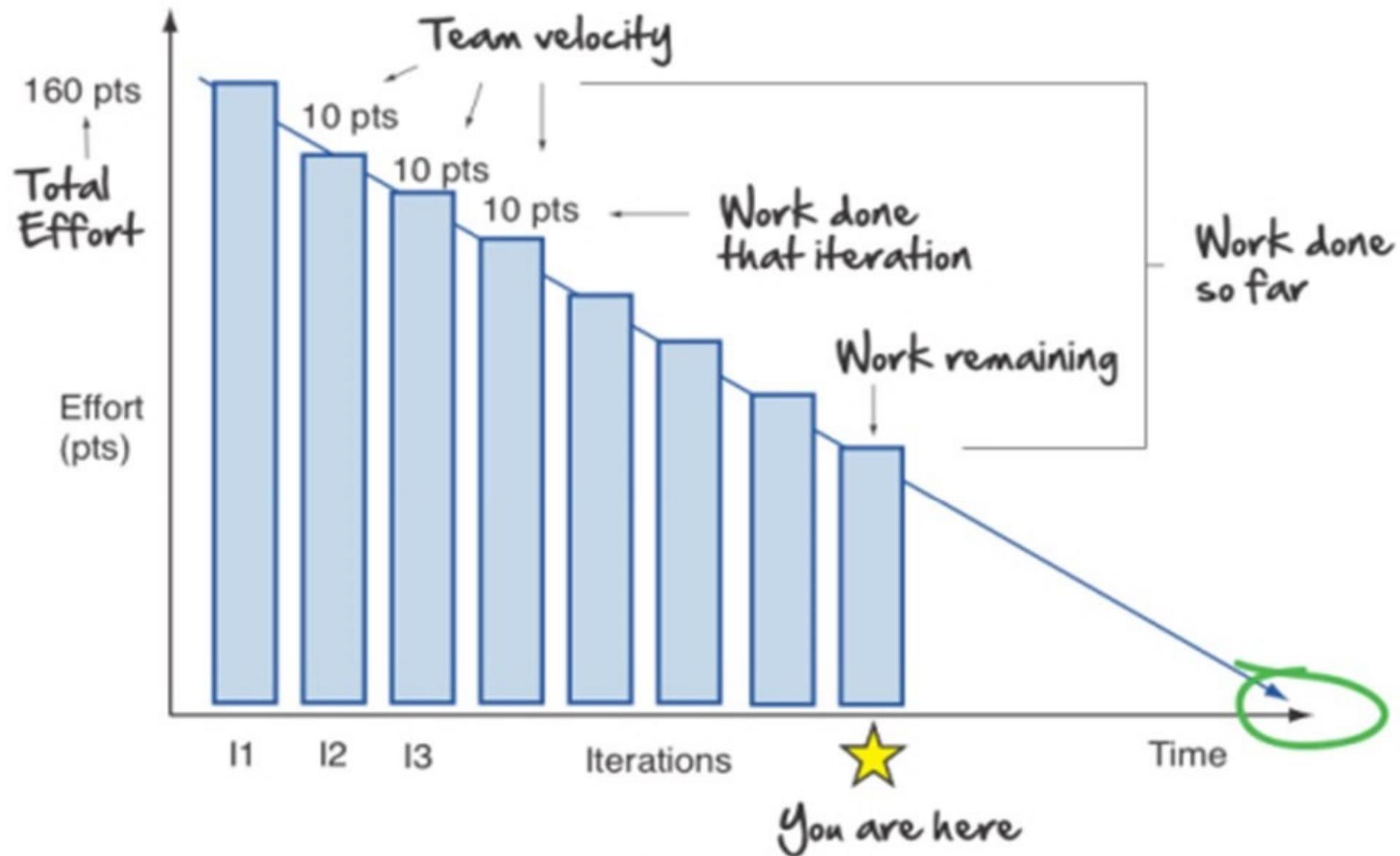


SCRUM - Burndown Charts

- ❑ A simple burndown chart is used in Scrum to show a trend of the remaining To Do estimate for all tasks within the sprint (iteration).
- ❑ By adding the completed effort to a burndown chart, the total current estimate within the iteration becomes visible as the total height of the To Do and Done bars.



SCRUM - Burndown Charts



Definition of Done

- Every Product Backlog item has acceptance criteria that define measurably what must be met when the item is declared to be done.
- Many criteria apply to all or many Product Backlog items.
- Instead of repeatedly defining these criteria with each item, it has proven to be useful to collect these criteria in one place: the Definition of Done.
- Thus, the Definition of Done is a shared understanding of the Scrum Team on the meaning of work to be complete. It typically contains quality criteria, constraints and overall non-functional requirements. Here is some examples:

Definition of Done

Definition of Done

- Reviewed by someone or a particular stakeholder
- Completed unit acceptance testing of the User Story
- Completion of quality assurance tests
- Completion of all documentation related to the User Story
- All issues are fixed
- Successful demonstration to stakeholders and/or business representatives

APPROVED

Roles



Product Owner:
Set priorities



ScrumMaster:
Manage process,
remove blocks



Team: Develop
product



Stakeholders:
observe & advise

Key Artifacts

Product Backlog

- List of requirements & issues
- Owned by Product Owner
- Anybody can add to it

Sprint Goal

- One-sentence summary
- Declared by Product Owner

Sprint Backlog

- List of tasks
- Owned by team

Blocks List

- List of blocks & unmade decisions
- Owned by ScrumMaster

Increment

- Version of the product
- Shippable functionality (tested,

Key Meetings

Sprint Planning Meeting

- Hosted by ScrumMaster; $\frac{1}{2}$ -1 day
- In: Product Backlog, existing product, business & technology conditions

 1. Select highest priority items in Product Backlog; declare Sprint Goal
 2. Team turns selected items into

Daily Scrum

- Hosted by ScrumMaster
- Attended by all, but Stakeholders don't speak
- Same time every day
- Answer: 1) What did you do yesterday? 2) What will you do today? 3) What's in your way?
- Team updates Sprint Backlog;

Sprint Review Meeting

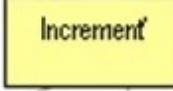
- Hosted by ScrumMaster
- Attended by all
- Informal, 4-hour, informational
- Team demos Increment
- All discuss
- Hold retrospective
- Announce next Sprint Planning

Development Process



Sprint Planning Meeting

30 days each



Scaling Scrum

SCRUM Estimation & Velocity

- "How many features will be completed?"
 - "When will we be done?"
 - "How much will this cost?"
-
- To answer these questions, we need to estimate the size of what we are building and measure the velocity at which we can get it done.

Relationship Among Size, Velocity, and Duration

- Basic Question: How much time do we need to create the features in Release 1?

- Answer:

Relationship Among Size, Velocity, and Duration

Estimated size ÷ measured velocity = (number of sprints)

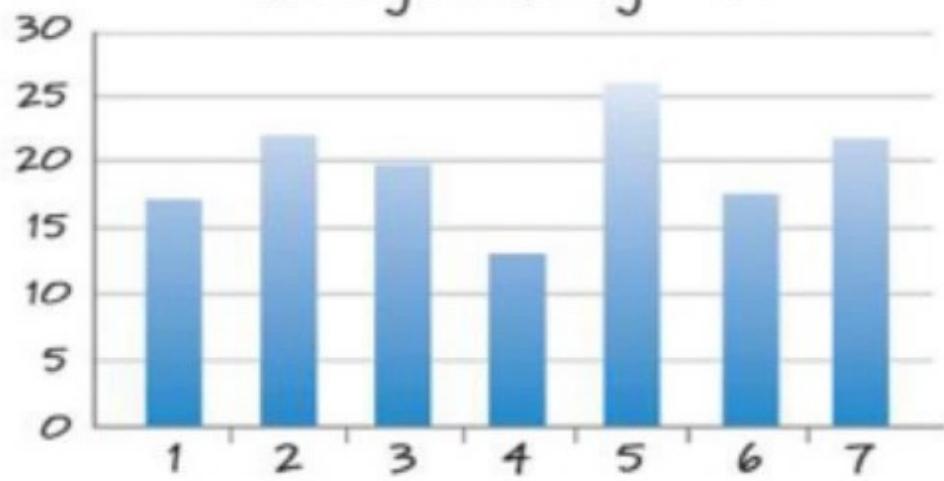
Item	Size
Feature A	5
Feature B	3
Feature C	2
Feature D	8
Feature E	2
Feature F	5
Feature G	3
Feature ...	1...
Feature ZX	5
Feature ZY	2
Feature ZZ	1
Feature ...	1...

$$200 \text{ points} \div 20 \text{ points/sprint} = 10 \text{ sprints}$$

$$\Sigma = 200 \text{ points}$$

Release 1

Average velocity = 20



Estimation: What and When

- Three different levels:

Portfolio backlog: Contains a prioritized list of all of the products (or projects) that need to be built.

- To estimate portfolio backlog items, rough, relative size estimates like Tshirt sizes are typically used.

Product backlog:

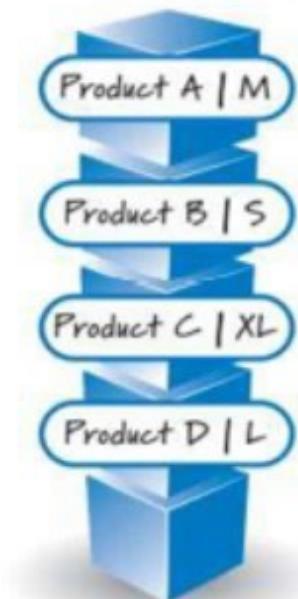
- To estimate coarse-grained PBIs, T-shirt sizes are typically used.

Sprint backlog:

- Development tasks are typically sized in ideal hours (Eg: effort-hours, man-hours, or person-hours).

Estimation: What and When

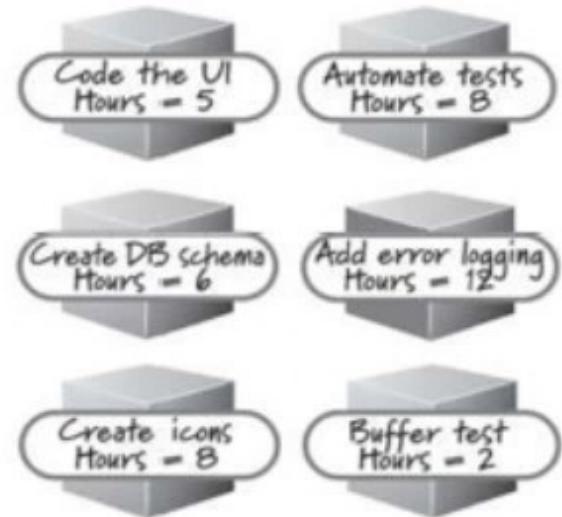
Portfolio backlog



Product backlog



Sprint backlog tasks



Item	Portfolio backlog	Product backlog	Sprint backlog tasks
Unit	T-shirt sizes	Story points / ideal days	Ideal hours / effort-hours
When	Portfolio planning	Product backlog grooming	Sprint planning

SCRUM Velocity & Capacity Relationship Calculation

Resource	No. of Days	Total Hours
A	10	65
B	8	52
C	5	33.5
D	10	65
E	9	58.5

Total Capacity: 241 hrs
Max. Capacity: 325 hrs

Velocity \propto Capacity



Story Points Productive Hrs.

Sprint-I	32	325
Sprint-II	38	325
Sprint-III	25	241
Sprint-IV	40	325
Sprint-V	? ₍₃₁₎	265
Sprint-VI	? ₍₃₂₎	273

Average

$$38/325 * 265 = 31$$

$$38/325 * 273 = 32$$

PBI Estimation Concepts

- 1. Estimate as a Team:** The development team that will do the work to design, build, and test the PBIs, will also do the estimation.
- 2. Estimates Are Not Commitments:** Our estimates should be accurate without being overly precise.
- 3. Accuracy versus Precision:**
- 4. Relative Size Estimation**

PBI Estimation Units

- **Story points:** Measure the bigness or magnitude of a PBI.
- **Ideal days:** Represent the number of effort-days or person-days needed to complete a PBI.
- **Technique:**
- **Participants:**

Scrum and Scalability

- Scrum: one of the **few agile methods** used to **scale up** for larger projects.
- How done?
 - Accomplished the same way as organizations handle integrated product teams.
 - Individual Scrum team coaches - part of a higher echelon team of coaches spanning several products.
 - This provides for communications to avoid conflicting development issues

Scrum - Queues

- **Product Backlog → Sprint Backlog → Sprint → Working increment of the Software**
- **Scrum** uses **lightweight queue-based management** and work-breakdown mechanisms.
- **Product Backlog queue:** a low-tech customer-managed queue of demand requests for products.
- **Sprint:** At launch time, a Sprint (30-day time-boxed iteration) does **just-in-time planning**
- **Sprint Backlog:** queue for Sprint work-mgmt.

Scrum - Management

- **Daily Scrum:** Very notable and very visible
- Is a **daily standup**,
 - **except** that it is the **team** that is participating and sharing coordination information **not a central project manager**.
- **Scrum Master**
 - holds daily scrum and
 - acts more as a **facilitator** and **runs interference** for the core team when **blocks** or **issues** arise.

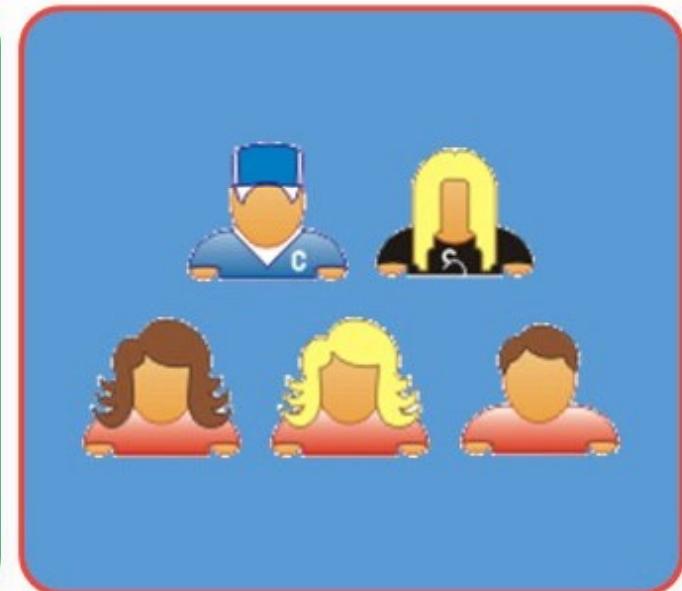
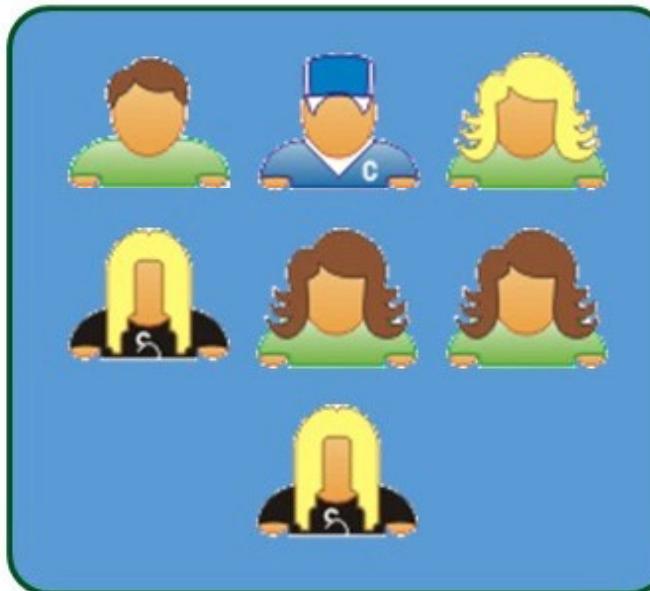
Scalability

- Typical individual team is 7 ± 2 people
 - Scalability comes from teams of teams

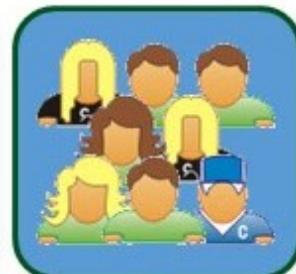
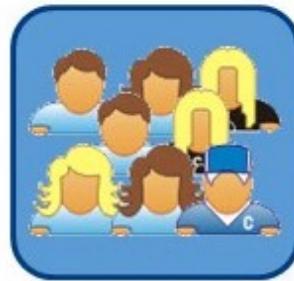
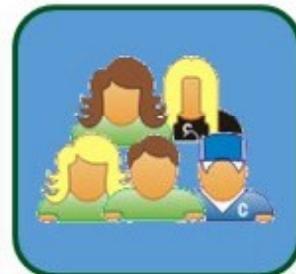
Factors in scaling

- Type of application
- Team size
- Team dispersion
- Project duration
- Scrum has been used on multiple 500+ person projects

Scaling through the Scrum of scrums



Scrum of scrums of scrums



Scrum Terminology

Scrum Team: Have already discussed

—**Product Owner:**

—**Scrum Master:**

—**Development Team:**

—**Product Backlog**

—**Sprint Backlog**

—**Sprint**

More Terminology Used in Scrum:

- **Sprint burn down chart:** Daily progress for a Sprint over the sprint's length.
- **(User) Story:** A feature added to the backlog is commonly referred to as a story; has a specific suggested structure.
- Done so development team can identify user, action and required result in a request; simple way of writing requests anyone can understand.
- **Example:** As a wiki user I want a tools menu on the edit screen so that I can easily apply font formatting.

More Terminology Used in Scrum:

- A **story** is an
 - independent,
 - negotiable,
 - valuable,
 - estimatable,
 - small,
 - testable requirement
- Despite being **independent**, stories have no direct dependencies with other requirements.
- Stories may be clustered into epics (a group of related stories) when represented on a product roadmap or further down in the backlog.

More Terminology Used in Scrum:

- **Tasks:** Added to story at beginning of a sprint and broken down into hours.
 - Each task should not exceed 12 hours, but it's common for teams to insist that a task take no more than a day to finish.
- **Definition of Done (DoD):** The **exit-criteria** used to determine whether a product backlog item is complete.
- In many cases the DoD requires that all **regression tests** should be successful.

More Terminology Used in Scrum:

- **Velocity:** The **total effort** a **team** is **capable of** in a **sprint**.
The number is derived by adding all the **story points** from the last sprint's stories/features.
- This is a **guideline** for the team and assists them in understanding **how many stories** they can do in a sprint.
- **Impediment:** Anything that prevents a team member from performing work as efficiently as possible.

References:

- www.mountaingoatsoftware.com/scrum
- www.scrumalliance.org
- www.controlchaos.com
- scrumdevelopment@yahooroups.com

References: A Scrum reading list

- Agile and Iterative Development: A Manager's Guide by Craig Larman
- Agile Estimating and Planning by Mike Cohn
- Agile Project Management with Scrum by Ken Schwaber
- Agile Retrospectives by Esther Derby and Diana Larsen

References: A Scrum reading list

- Agile Software Development Ecosystems by Jim Highsmith
- Agile Software Development with Scrum by Ken Schwaber and Mike Beedle
- Scrum and The Enterprise by Ken Schwaber
- Succeeding with Agile by Mike Cohn
- User Stories Applied for Agile Software Development by Mike Cohn