



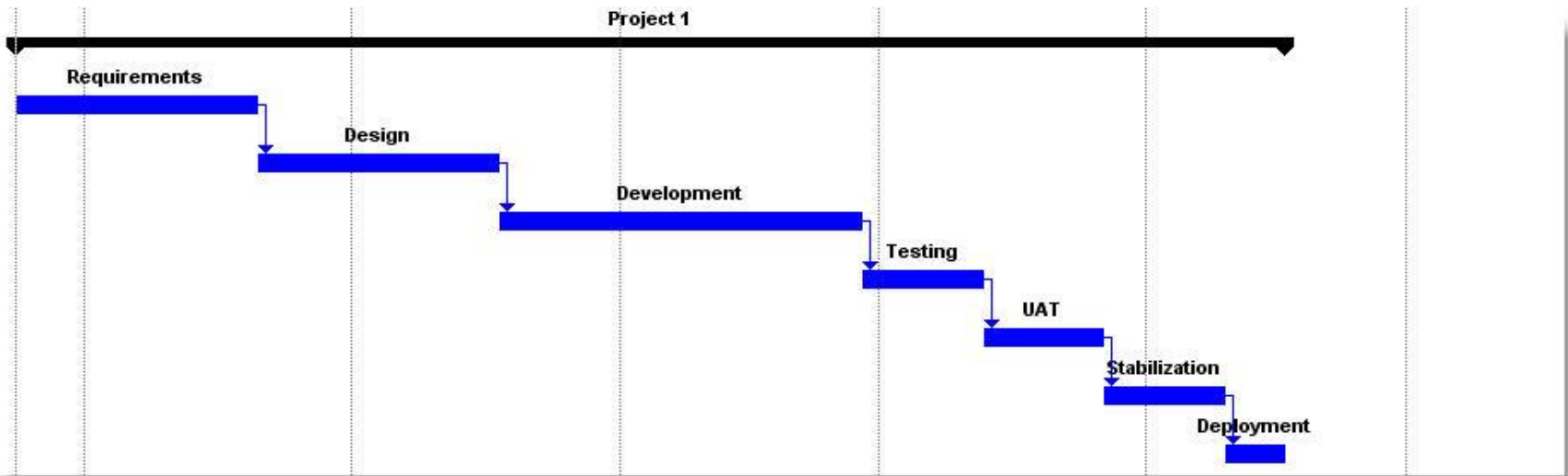
Introduction to Agile Planning and Estimation

*Lee Cunningham, CSM, CSP, PMP
VersionOne Agile Coach & Product Consultant*

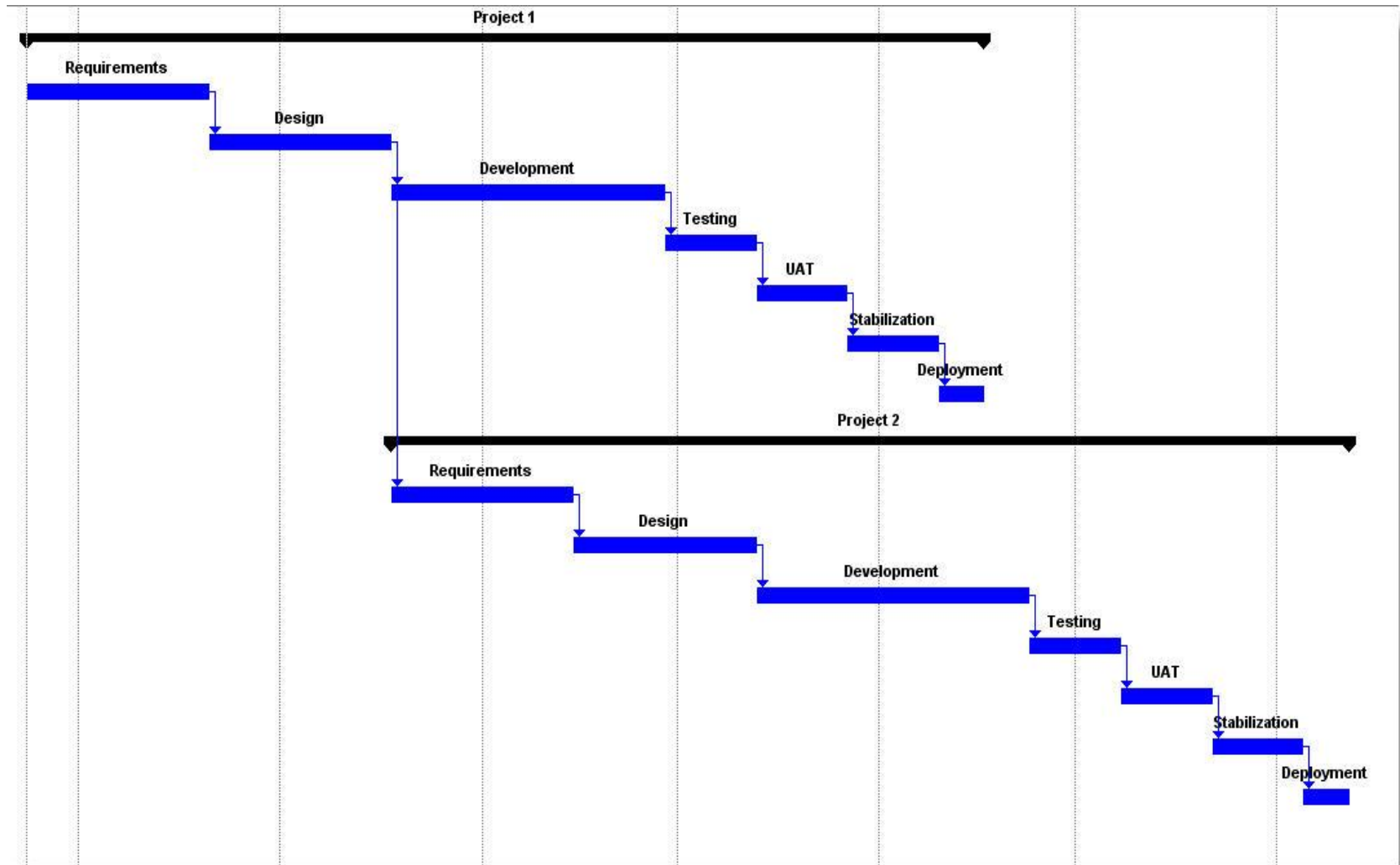
Goals for Today

- Learn about the Agile planning and estimation mindset (why)
- Learn some agile planning and estimation techniques (how)
- Have fun

A Familiar Pattern



Making Things Better(?)



Consider...

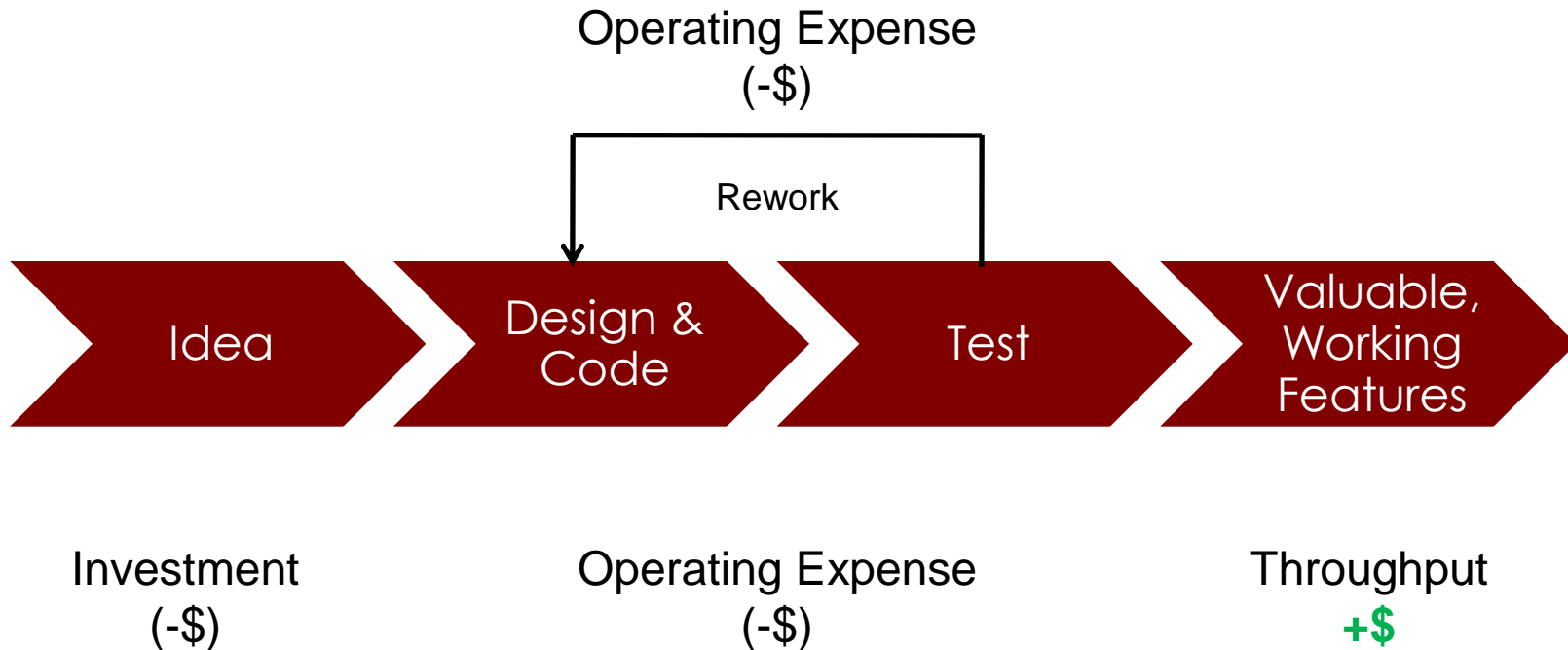
It is possible to finish
on schedule
and under budget,
but still not deliver
anything of value.

Earned Value

Ordering a cake from a bakery...



Throughput Accounting



Maximize Throughput by removing system constraints while limiting Investment and Operating Expenses.

Agile Software Development

- Agile software development is a **conceptual framework** for software engineering that promotes development iterations throughout the life-cycle of the project.
- Agile **minimizes risk** by developing software in short amounts of time. Software developed during one unit of time is referred to as a sprint, which may last from one to four weeks.
- **Each sprint is an entire software project:** including planning, requirements analysis, design, coding, testing, and documentation. A sprint may not add enough functionality to warrant releasing the product to market but the **goal is to have an available release (without bugs) at the end of each iteration.**
- **At the end of each sprint, the team re-evaluates project priorities.**

http://en.wikipedia.org/wiki/Agile_software_development

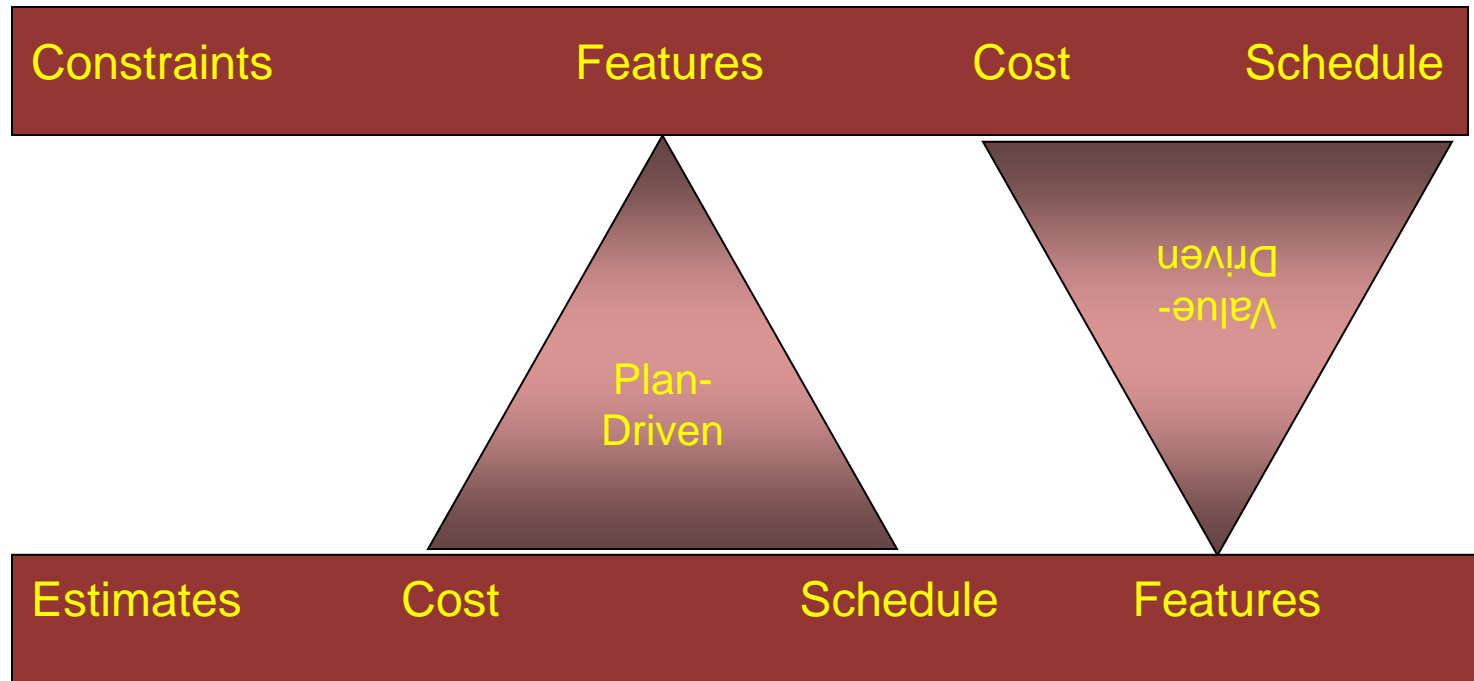
What do the various agile methods have in common?

- **Customer satisfaction** by rapid, continuous delivery of useful software
- **Working software** is delivered frequently (weeks rather than months)
- Working software is the **principal measure of progress**
- Even late **changes in requirements are welcomed**
- Close, daily **cooperation between business people and developers**
- **Face-to-face** conversation is the best form of communication
- Projects are built around **motivated individuals**, who should be trusted
- Continuous attention to **technical excellence** and good design
- **Simplicity**
- **Self-organizing teams**
- **Regular adaptation** to changing circumstances

Lean Principles

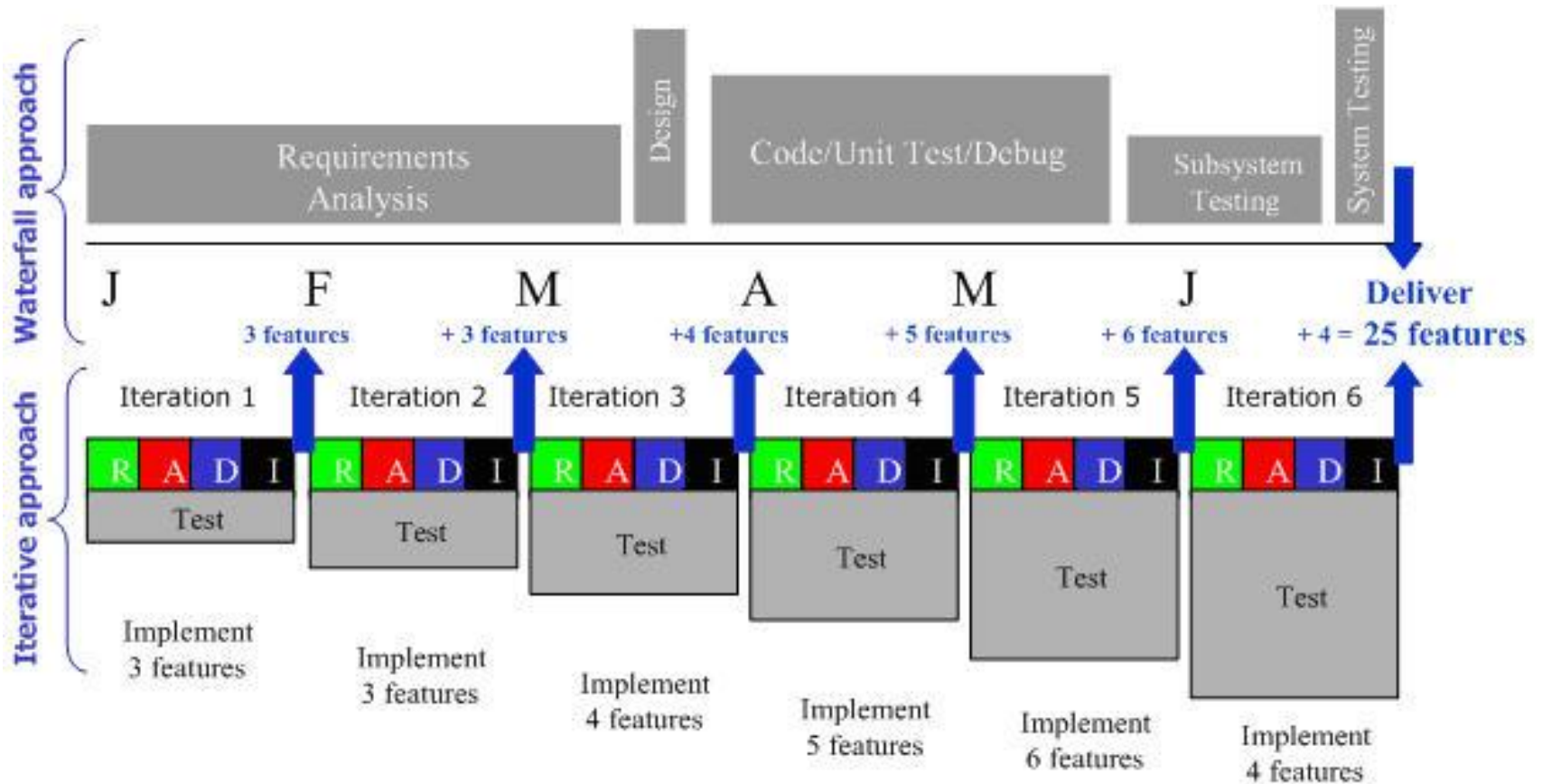
- **Higher Quality:** “Designed-to-fit” product with flexibility to change.
- **Increased Throughput:** Iterative and incremental project and product “chunks” with earlier value delivery.
- **Reduced Waste:** Lean, efficient processes with lower costs and higher productivity.
- **“Measure Up”:** Fewer, but more meaningful measures

The “Iron Triangle”

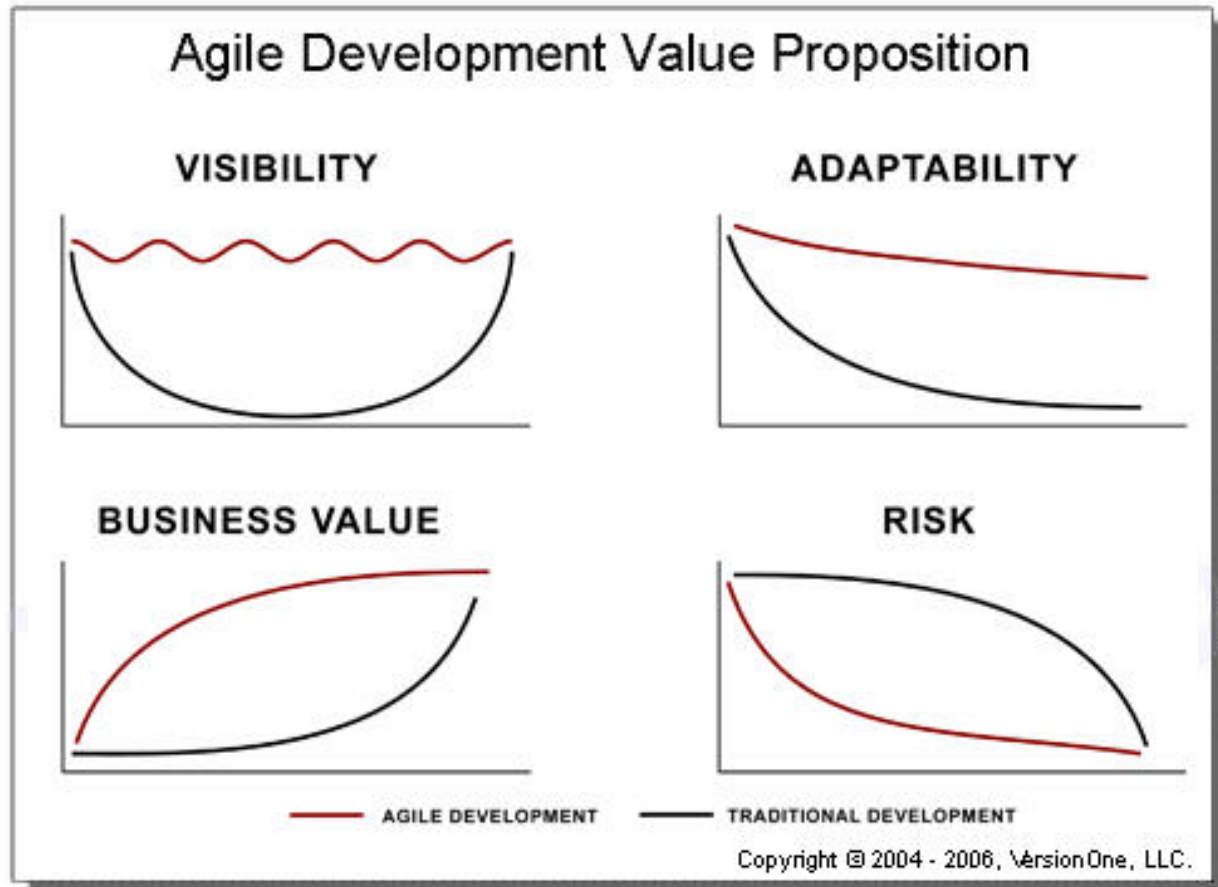


- Plan-driven methodologies generally constrain features and vary time and cost
- Agile methodologies generally constrain time and cost and vary scope

Waterfall vs. Agile Planning



What do You Think?



Backlogs

Backlog Fundamentals

Backlog

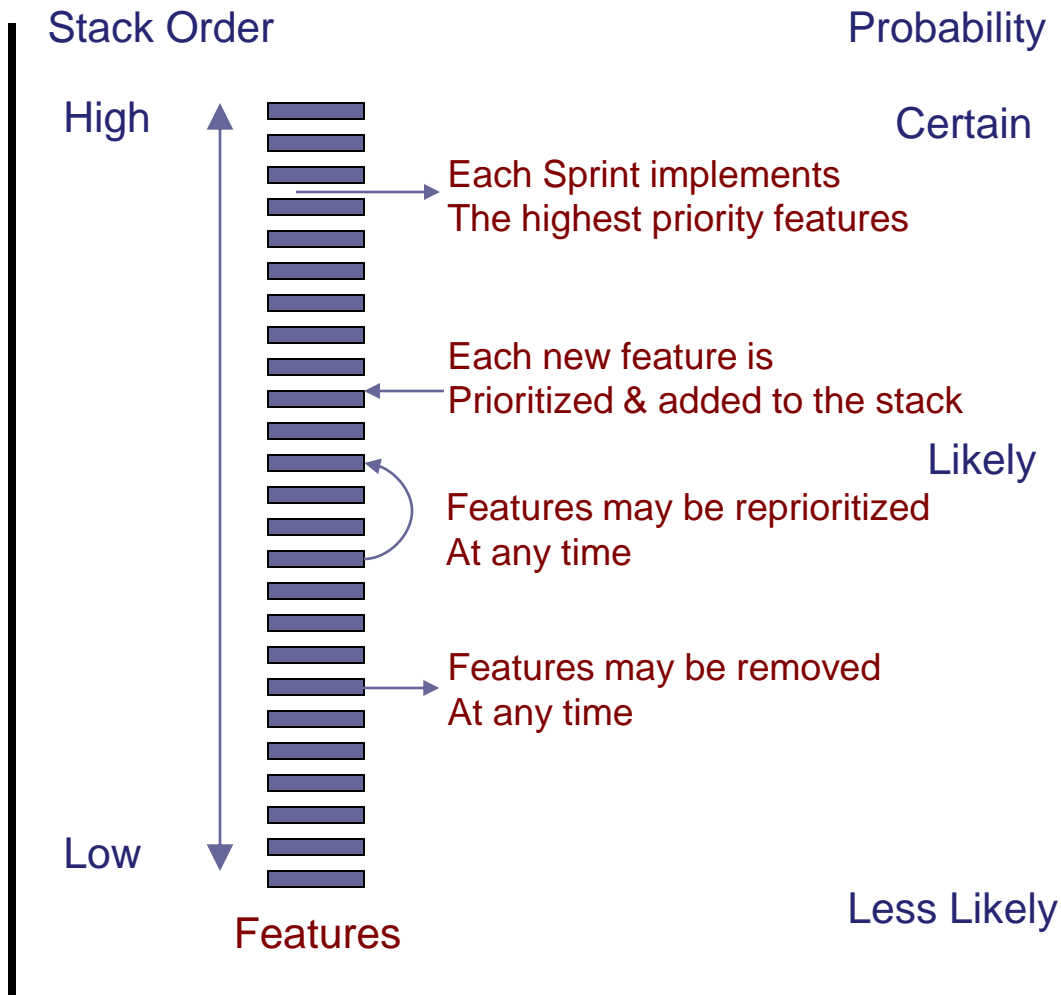
A prioritized list

Backlog item (Story)

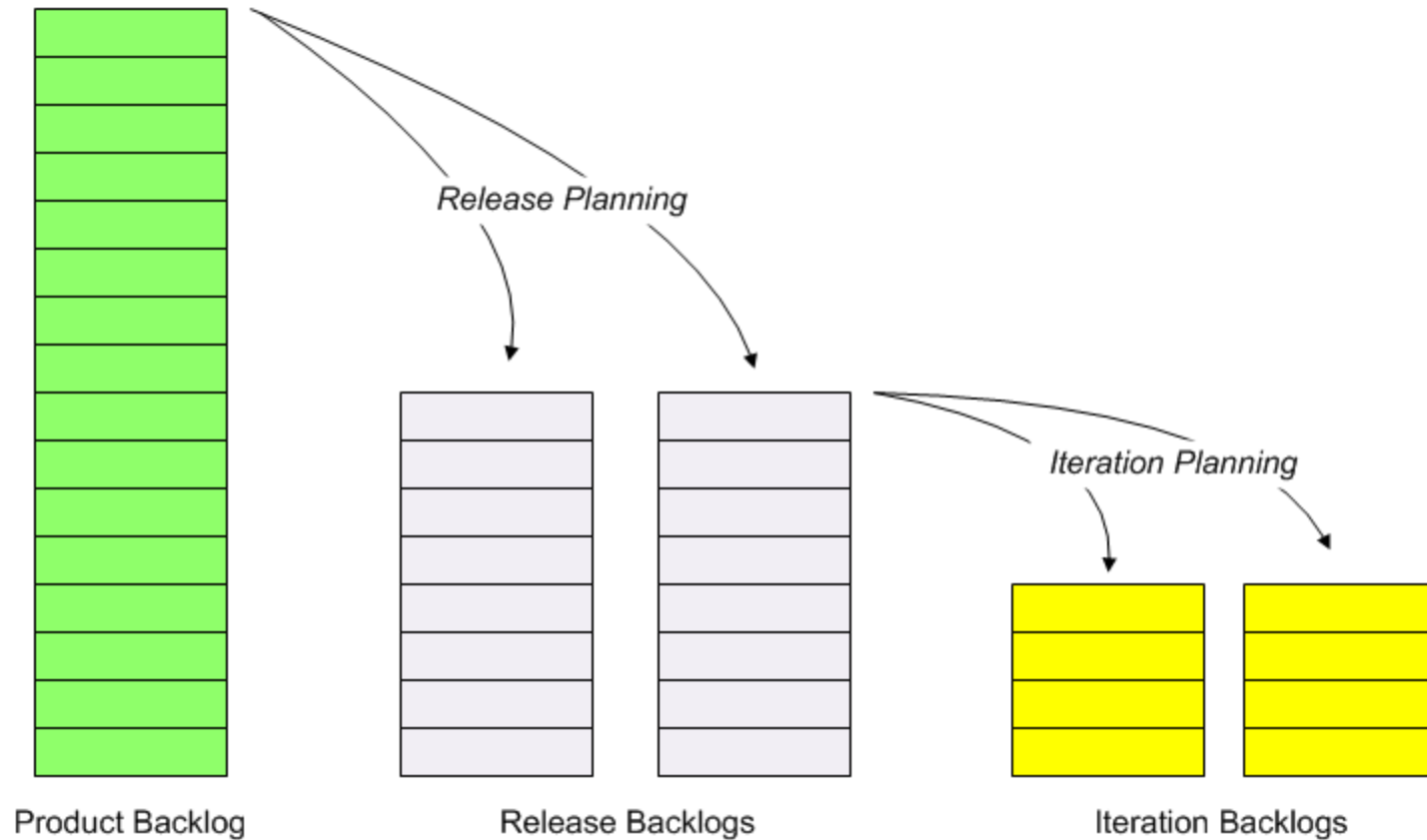
A high level description of an individual feature in the backlog. It is a placeholder for a future conversation about that feature.

Product Backlog Design

- All possible system features are captured in a prioritized list – the Product Backlog
- New features can be added at any time to the Product Backlog
- Features have only a gross estimate of effort and value



Planning and Backlogs



Backlog Item Prioritization

- Backlogs are lists of stories that are arranged in rank order – “What is the next most important thing to do?”
- The order in which the stories are arranged depends on the combination of factors that are important to your business
- These individual factors can (and will) change, which will affect the rank order of the stories
- The Product owner ultimately decides the rank order

Agile Estimation

Time vs. Relative Complexity

Room painting example...

We're better at sizing things up than we are determining how long they'll take.



Planning Poker



Velocity

Velocity

- The **rate** at which a team can produce working software
- Measured in non-time-referent terms (e.g., Story Points) per Sprint
- More accurately stated, it is measured in terms of the **stabilized number** of Story Points a team can deliver **per sprint of a given length, and with a given definition of Done.**
- Used for estimation and planning
- Can be artificially increased by cutting corners on quality
- Must have stabilized to be reliable
- Should not be used as a measure of comparison across teams
- Lean concept of Limiting Work to Capacity

Velocity

Example: Team A is working in 2-week sprints on work that it has estimated together. Team A has been working together for several sprints, and consistently delivers between 18 and 23 points of working software every sprint.

- We could reasonably expect Team A to deliver roughly 20 points per 2-week sprint, and so we consider that to be the team's velocity for planning purposes.
- If there are eight 2-week sprints in a release, we can extrapolate that Team A has the capability to deliver 160 points in a release.

Connecting the Dots

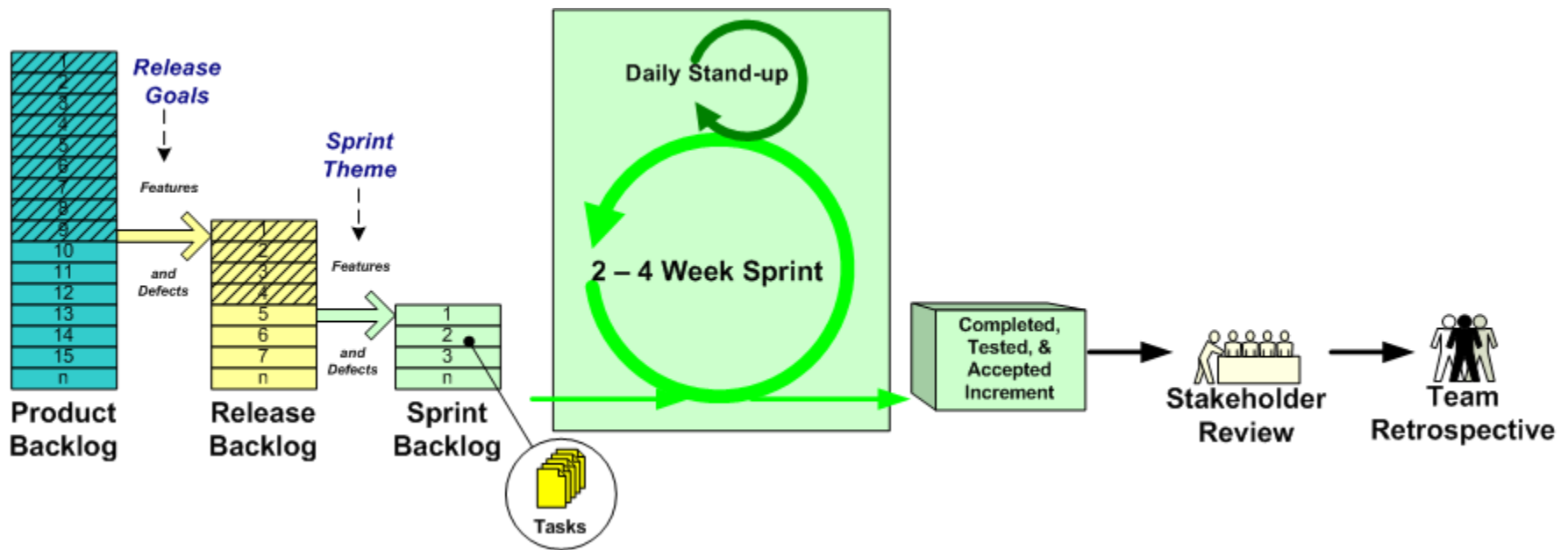
- **Size** (complexity) is estimated
 - A story is estimated to be 3 story points in relative complexity
- **Velocity** is measured
 - “Team A can deliver 20 story points in a 3-week sprint”
- **Duration** is derived
 - “Based on Team A’s measured velocity of 20 story points per sprint, it will take Team A 3 sprints to deliver 60 story points.”

So...

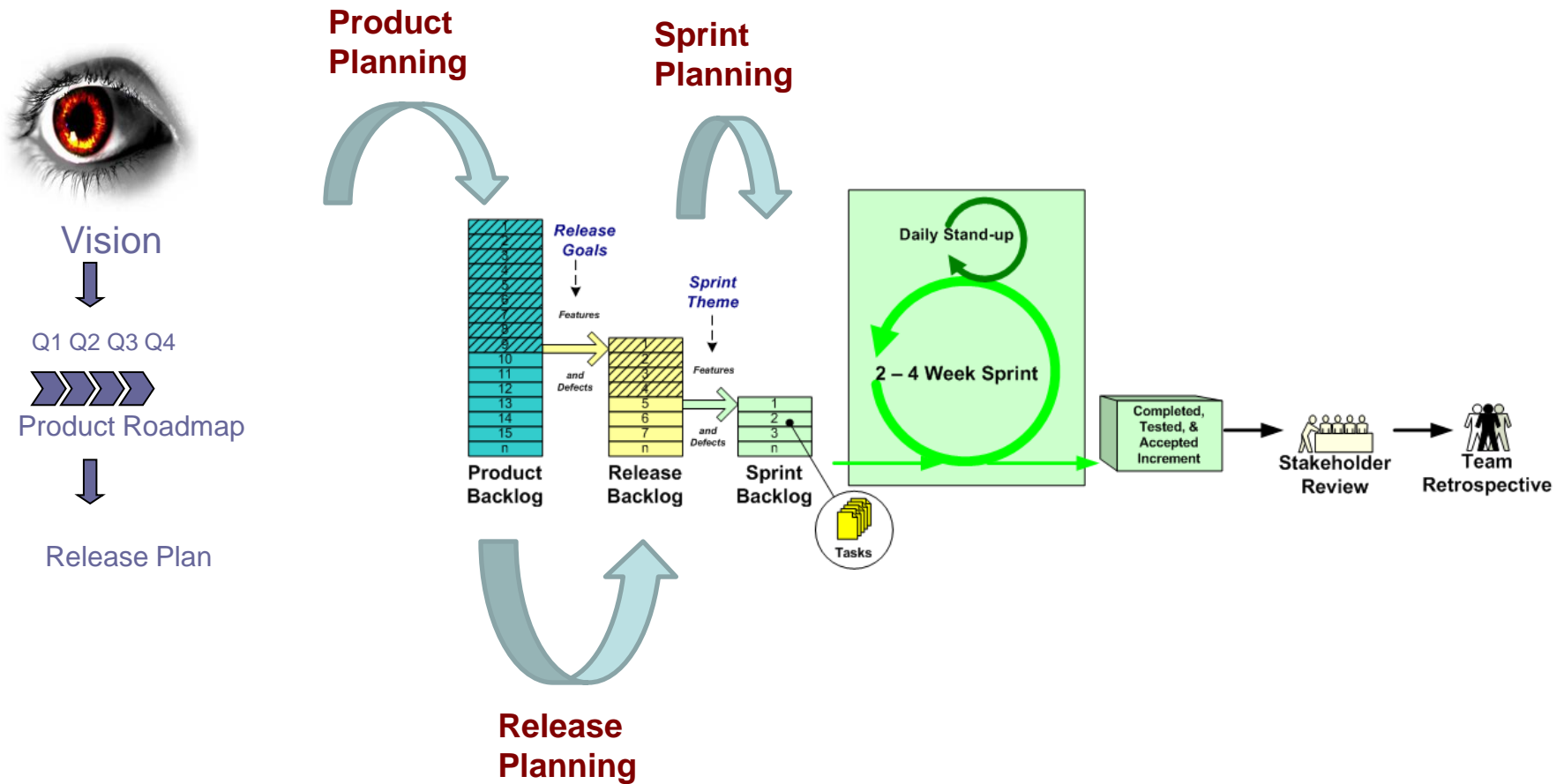
- Backlog Item estimates answer the question “how big?”, rather than “how long?”
- Size estimates and observed Velocity, used together, are answer the question “how long?”

Agile Planning

Scrum Framework

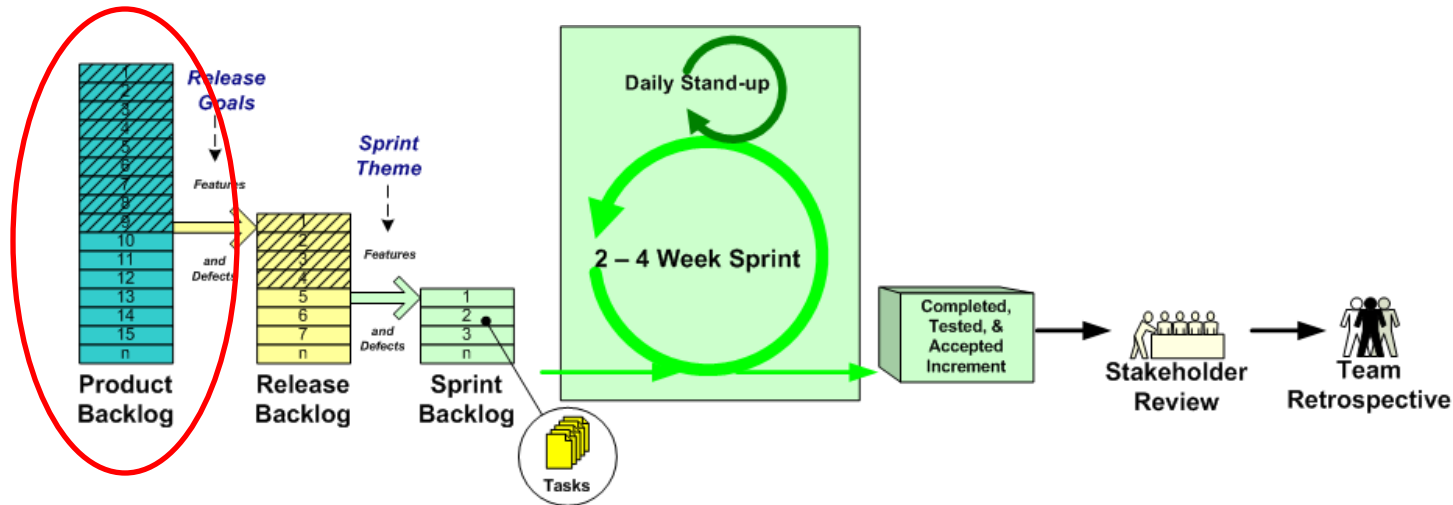


Expanded Framework



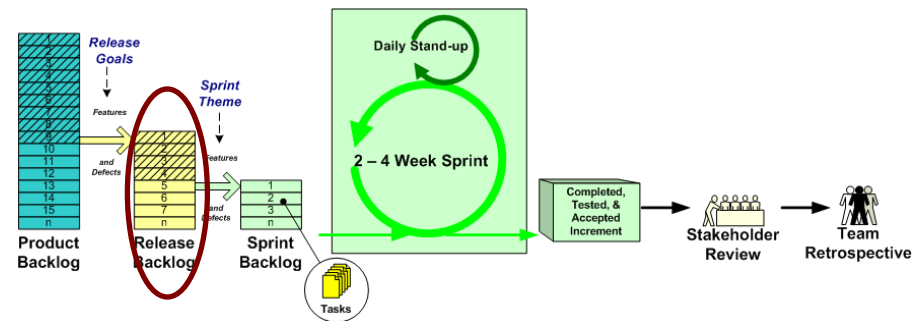
Product Planning

- Answers the question, “What business objectives will the product fulfill?”
- Is ongoing during the life of the product
- Results in the Product Backlog



Release Planning

- Answers the questions,
 - How can we release value incrementally?”
 - What subset of business objectives will each release achieve?
 - What user constituencies will the release serve?
 - What general capabilities (epics, feature groups) will the release offer?
- Results in the
 - Release Roadmap
 - Release Plan
 - Release Backlog



Release Planning

Release Planning :: Call Center

Release Scheduling | Release Forecasting | Team Scheduling

1/14/2010 **Call Center**
Total Estimate - Rollup: 322.00

Release 1.0 (1/14/2010 - 8/11/2010) | Release 2.0 (8/14/2010 - 5/14/2011) | Release 3.0 (1/14/2010 - 5/14/2011)

Release	Start Date	End Date	Estimate	Backlog Item Count	Backlog Item Estimate	Defect Count	Defect Estimate
Release 1.0	1/14/2010	8/11/2010	192.00	17	67.00	13	28.00
Release 2.0	8/14/2010	5/14/2011	5.00	1	5.00	0	0
Release 3.0	1/14/2010	5/14/2011	5.00	0	0	0	0

Backlog Items / Defects

Filter: Project: Call Center

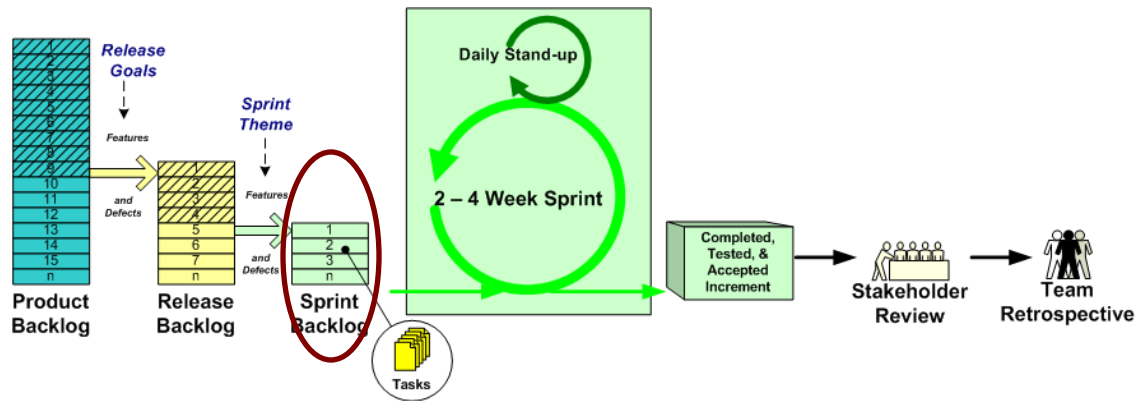
Move to Project: 1-10 of 10

Title	Feature Group	Priority
Partial Shipments Doubled		Low
Warehouse Integration - Shipping	Integration	High

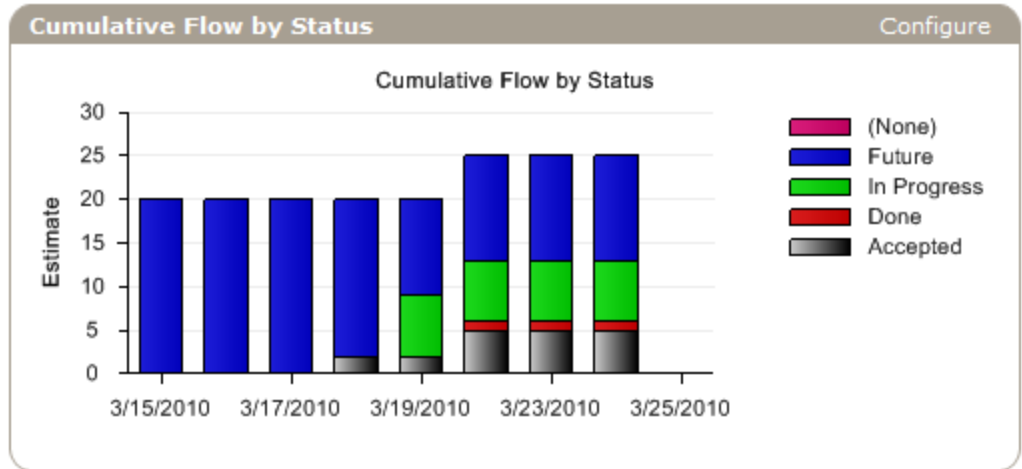
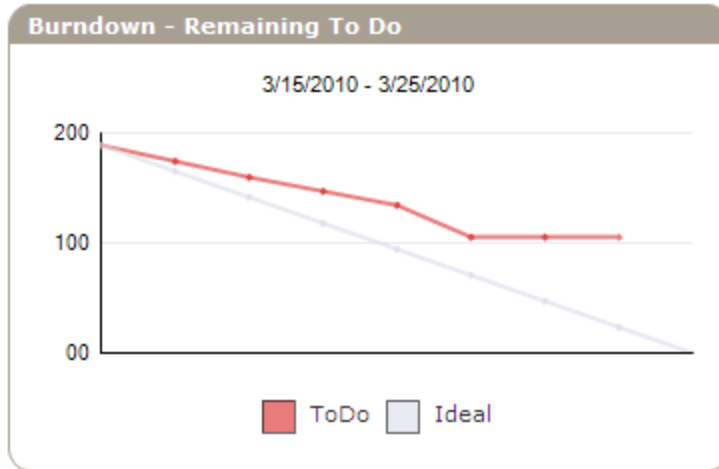
Sprint Planning

Sprint Planning

- Answers the questions, “What are we building over the next 2-4 weeks?”
- Results in the Sprint Backlog



Focus on Results



How much work do we have left to do?

How much value have we delivered?

Effort Expended does not equal Value

Questions?



Thanks!

