

AGILE Realities

Helping our customers to
realize their business ideas



Presenters:

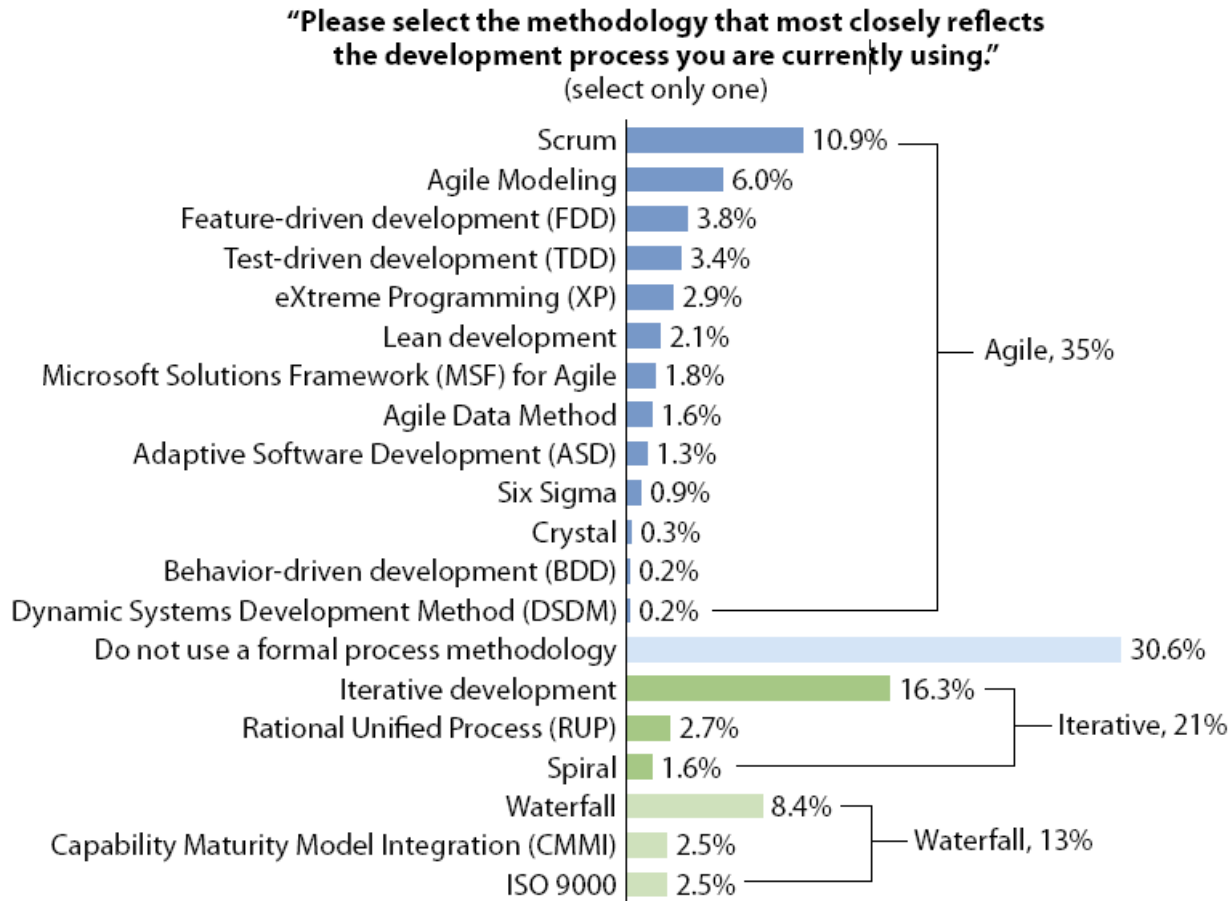
Chris Koo (Edward Jones)

Blake Moyer (Edward Jones)

Joan Romine (Boeing)

AGILE Gaining Momentum

Figure 1 Agile Is Organizations' Primary Development Approach



Base: 1,298 IT professionals

Source: Forrester/Dr. Dobb's Global Developer Technographics® Survey, Q3 2009

AGILE First Impressions



Fast!



Flexible!



AGILE Development Myths

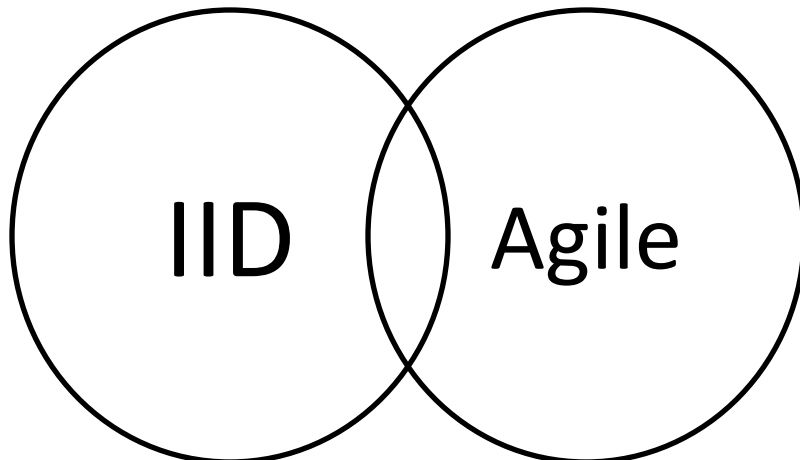
- ✗ "Agile teams don't plan."
- ✗ "Agile is a silver bullet."
- ✗ "Agile teams don't need requirements."
- ✗ "Agile teams can deliver an end product faster than waterfall."

AGILE Development

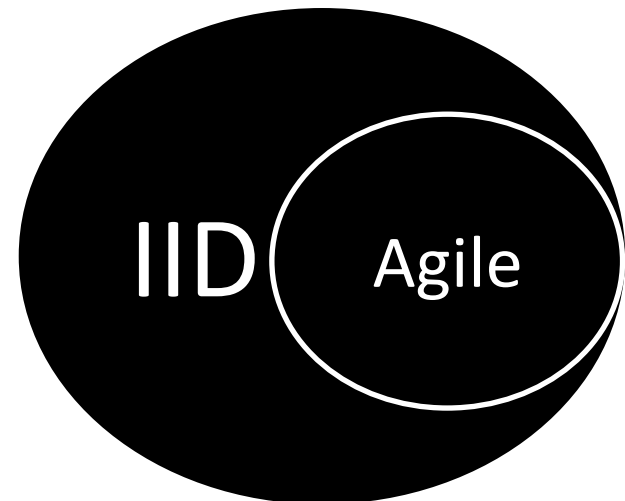
Is iterative and incremental

- Is about useful working software quicker and better through better internal team and customer collaboration.
- Is a mindset and discipline, NOT a template or tool.

You may think that IID and Agile work together like this...



A better representation of IID and Agile would be as follows...



Manifesto for AGILE Development

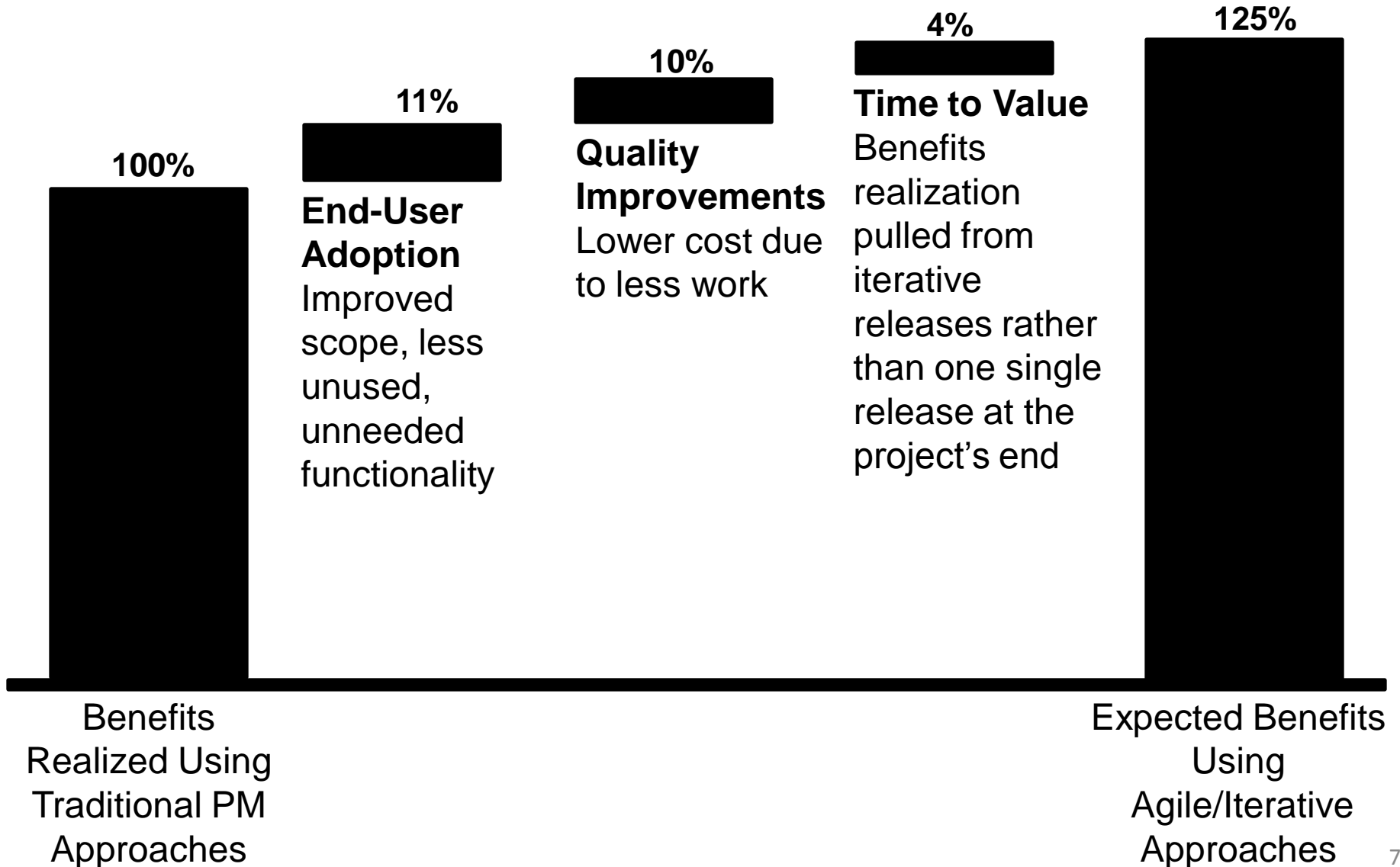
We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

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

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

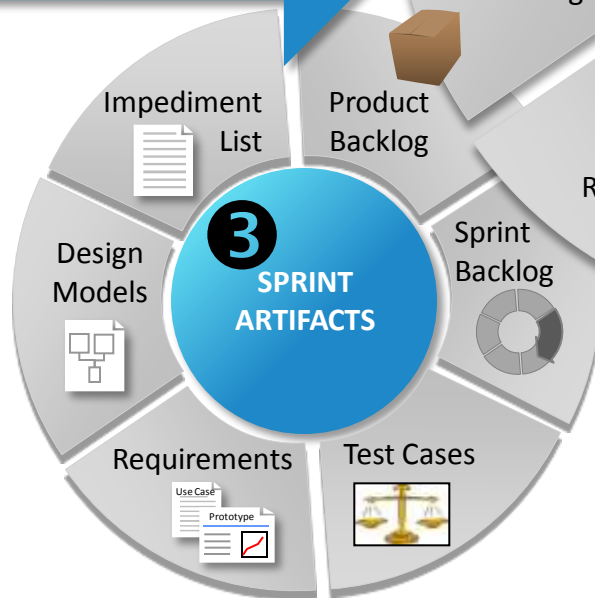
AGILE Project Returns



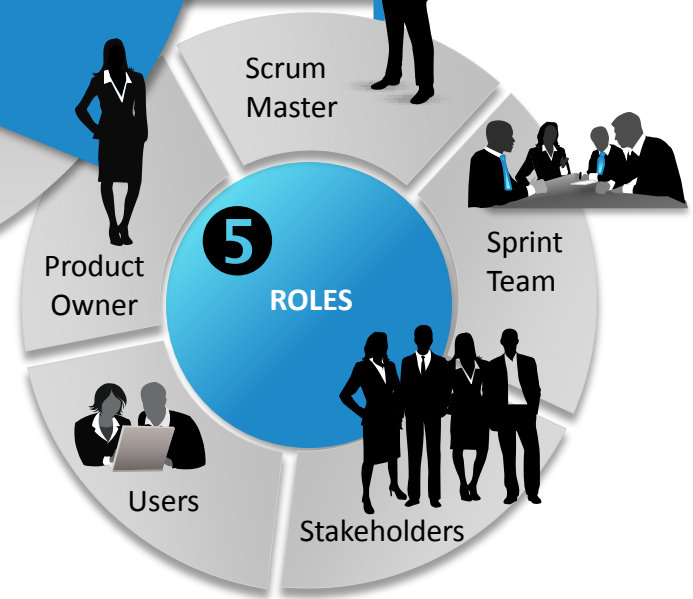
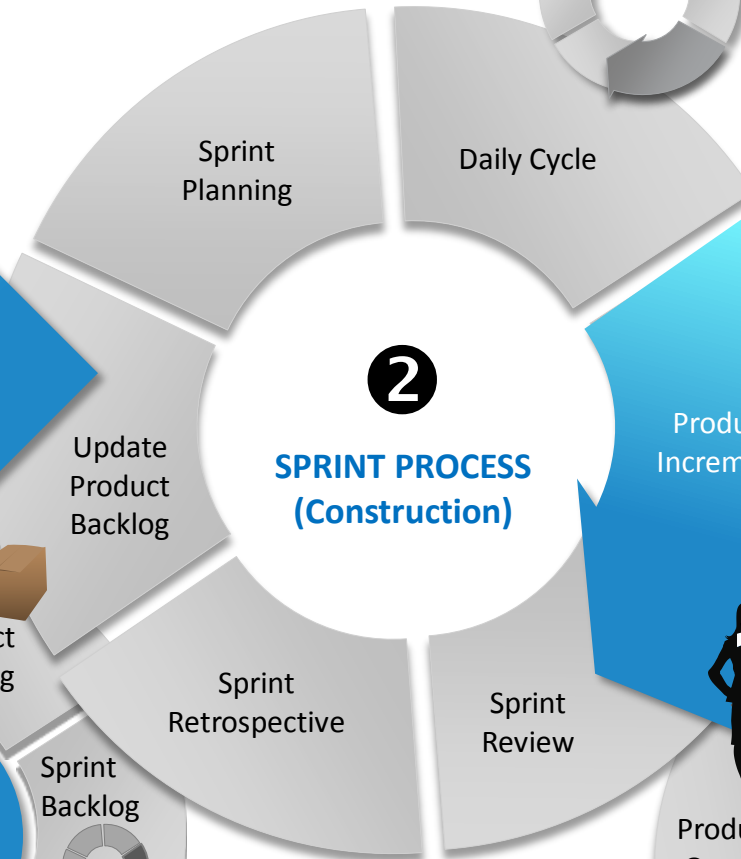
AGILE Process

1 PREPARATION (Iteration -1 & 0)

- Vision / Business case
- Agile eligibility / approval
- Epic requirements
- Initial product backlog
- Initial release plan
- Team assembly

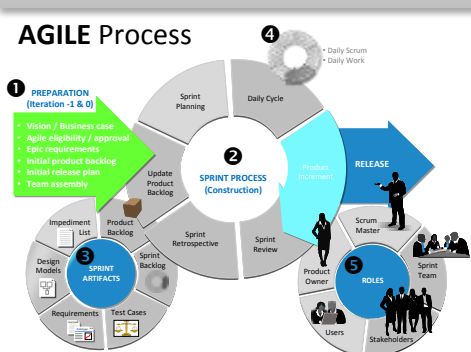


2 SPRINT PROCESS (Construction)



① AGILE – Preparation

- Vision / Business case
- Agile eligibility / approval
- EPIC requirements
- Initial product backlog
- Initial release plan
- Team assembly



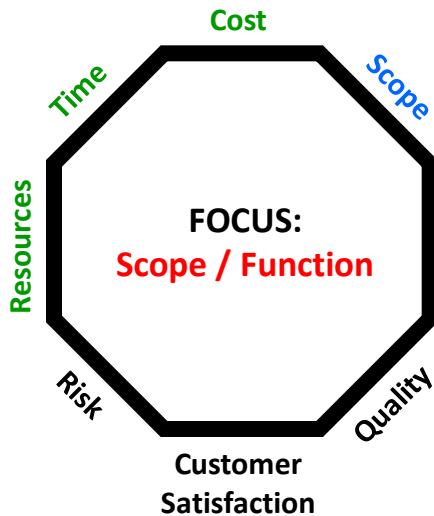
Mindset Shift - Scope



#113 - "AGILE DEVELOPMENT, EXPLAINED" - BY SALVATORE IOVENE, FEB. 21ST 2009

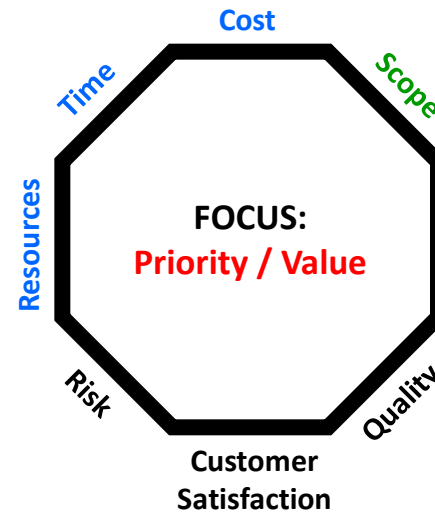
[HTTP://WWW.GEEKHEROCOMIC.COM/](http://www.geekherocomic.com/)

Waterfall



- Fixed Constraint
- Flexible Constraint

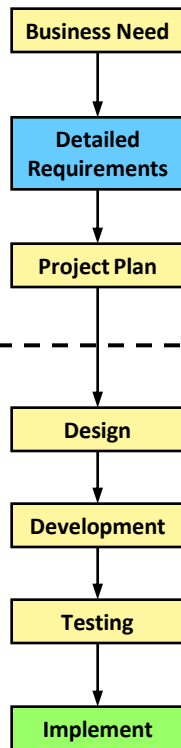
AGILE



Mindset Shift - Requirements

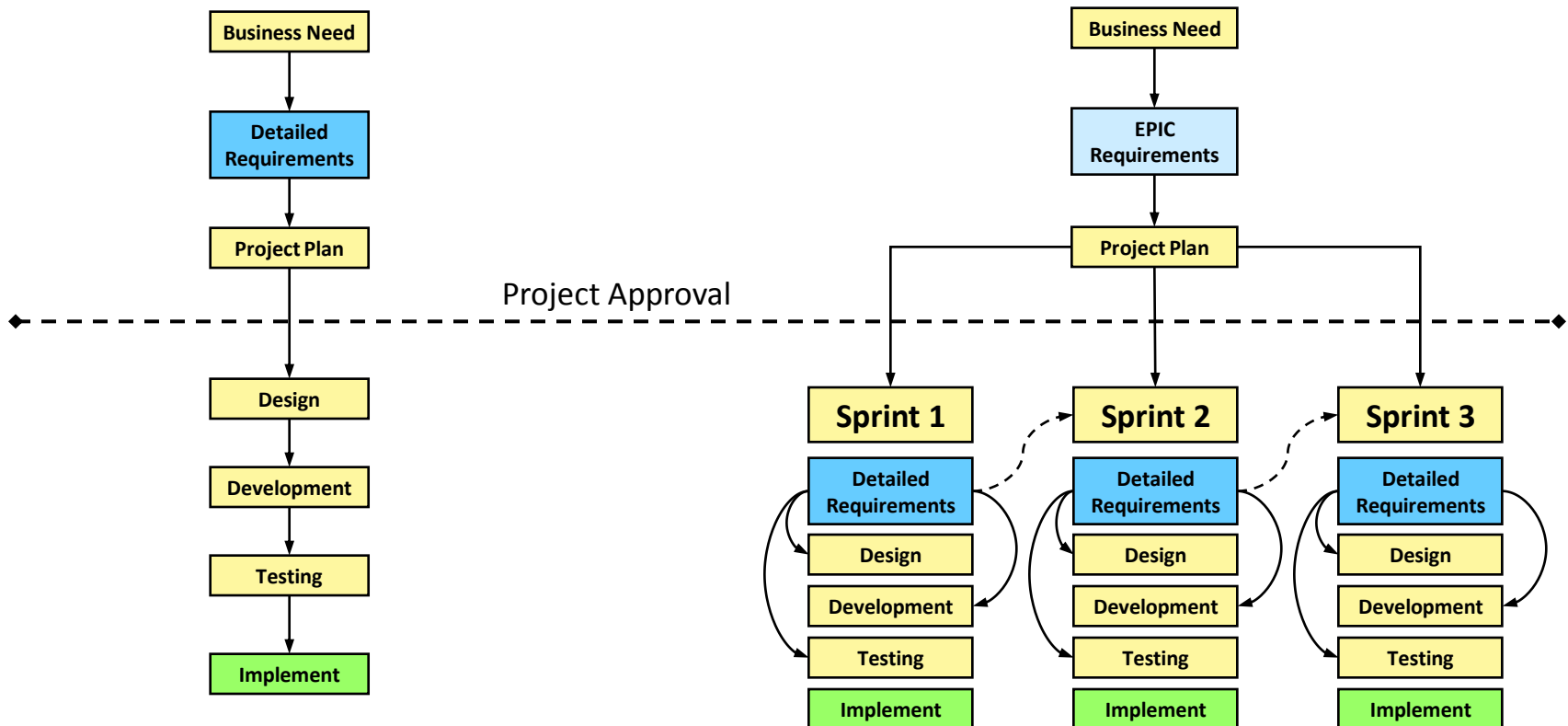
Waterfall

"Pay now..."



AGILE

"...or pay later"



Mindset Shift – Product Backlog

Business-defined story or use case describing a function.

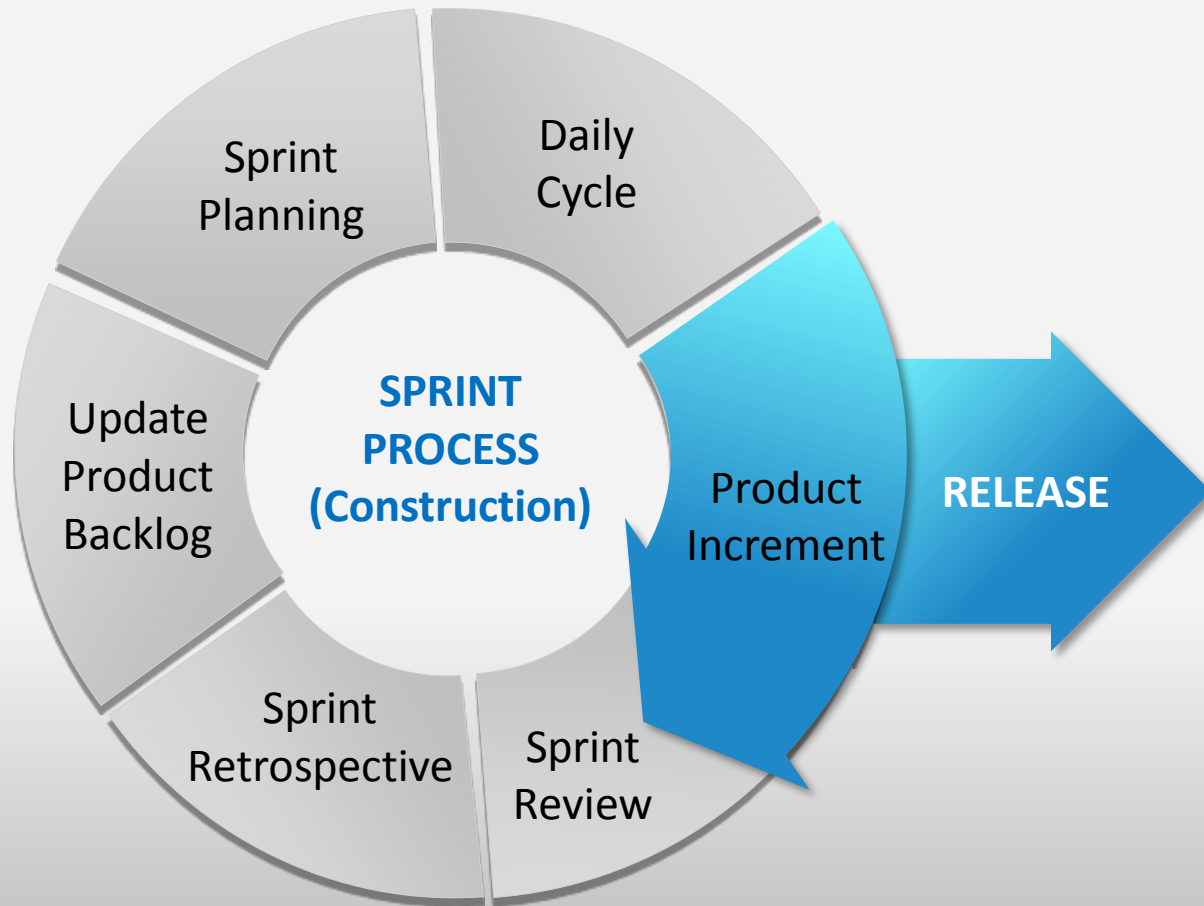
Highest priority stories are developed first; others queued up as "Planned."

Business defines story priorities, and may change them at any time

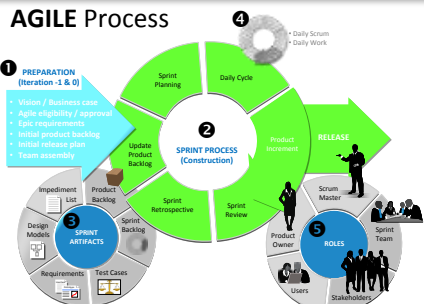
Sprint team sizes stories using a relative measure; at the end of each sprint, # story pts delivered is measured.

Story ID	Story Name / Description	Status	Business Priority	Story Points	Sprint
1	View Orders (INQ Summary)	Planned	A		1
2	INQ Summary Screen (Branch users only)	Planned	A.0		1
3	View Orders by Name / Account - Includes creation of View file	Planned	A.1	7	1
4	View Orders by FA	Planned	A.7	3	1
5	View Orders by Symbol	Planned	A.2	2	1
6	View Orders by Product	Planned	A.3	3	1

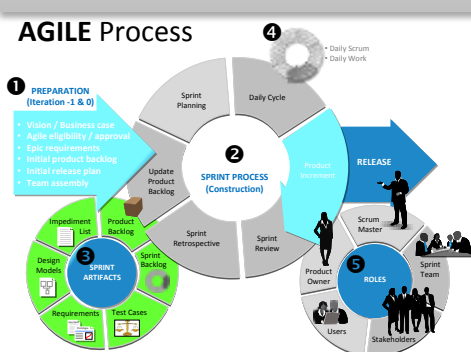
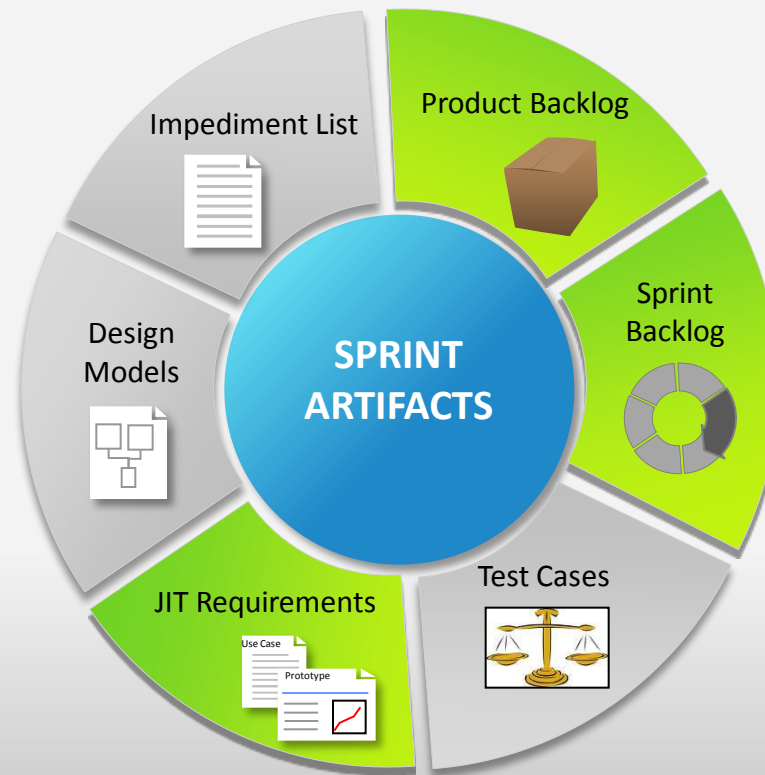
② AGILE – Sprint Process



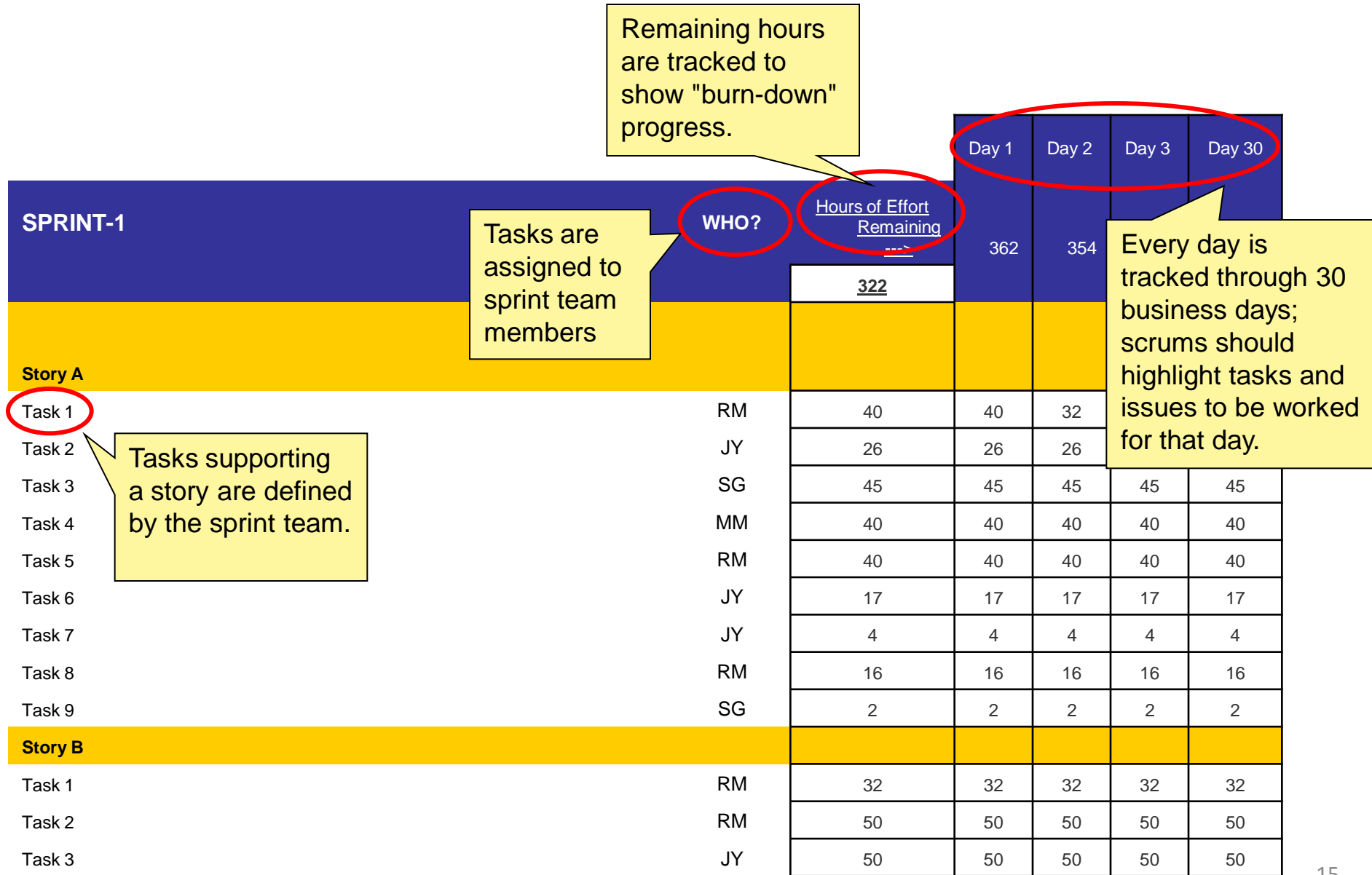
AGILE Process



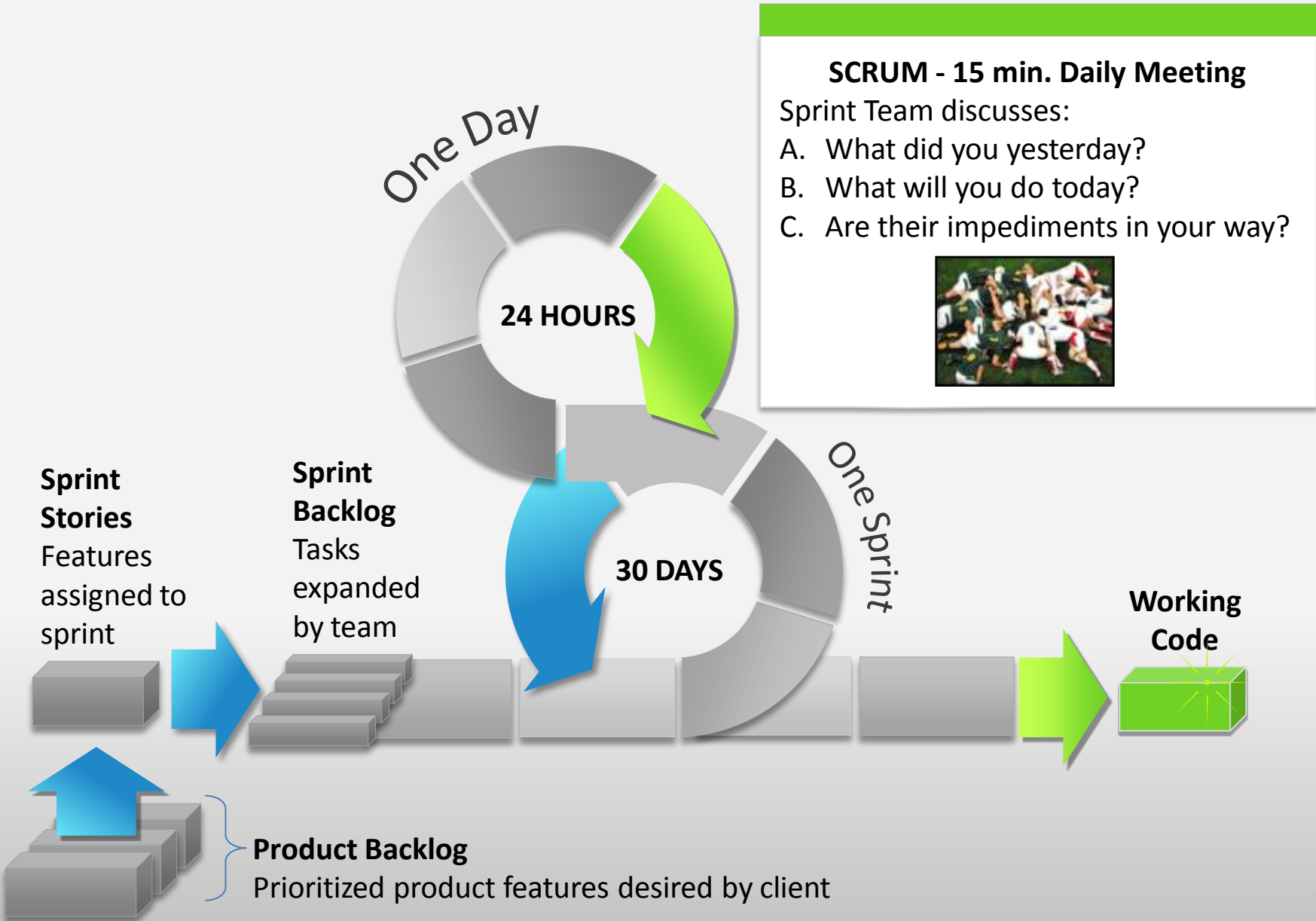
③ AGILE - Sprint Artifacts



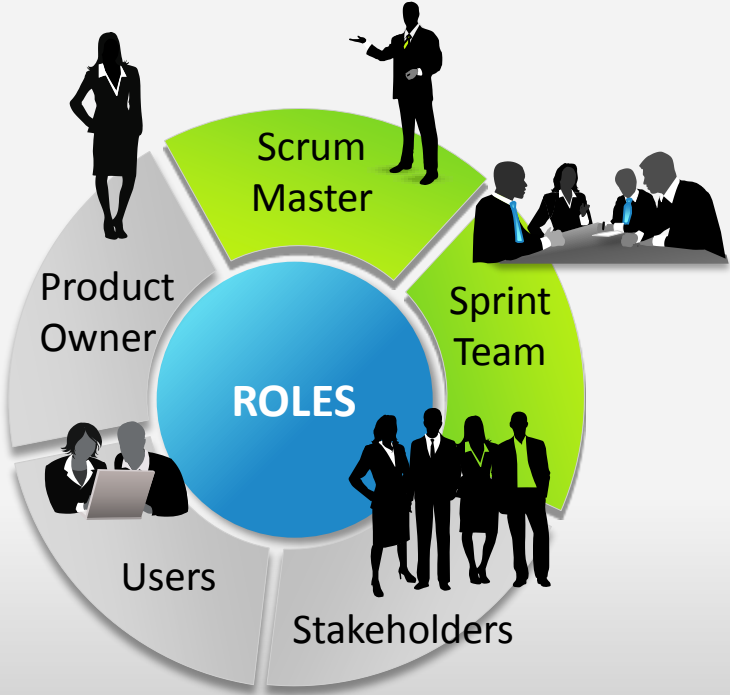
Mindset Shift – Sprint Backlog



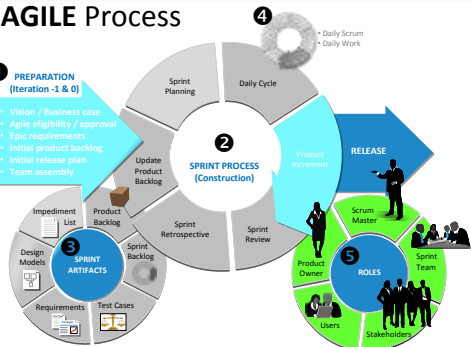
④ AGILE – Daily Scrum



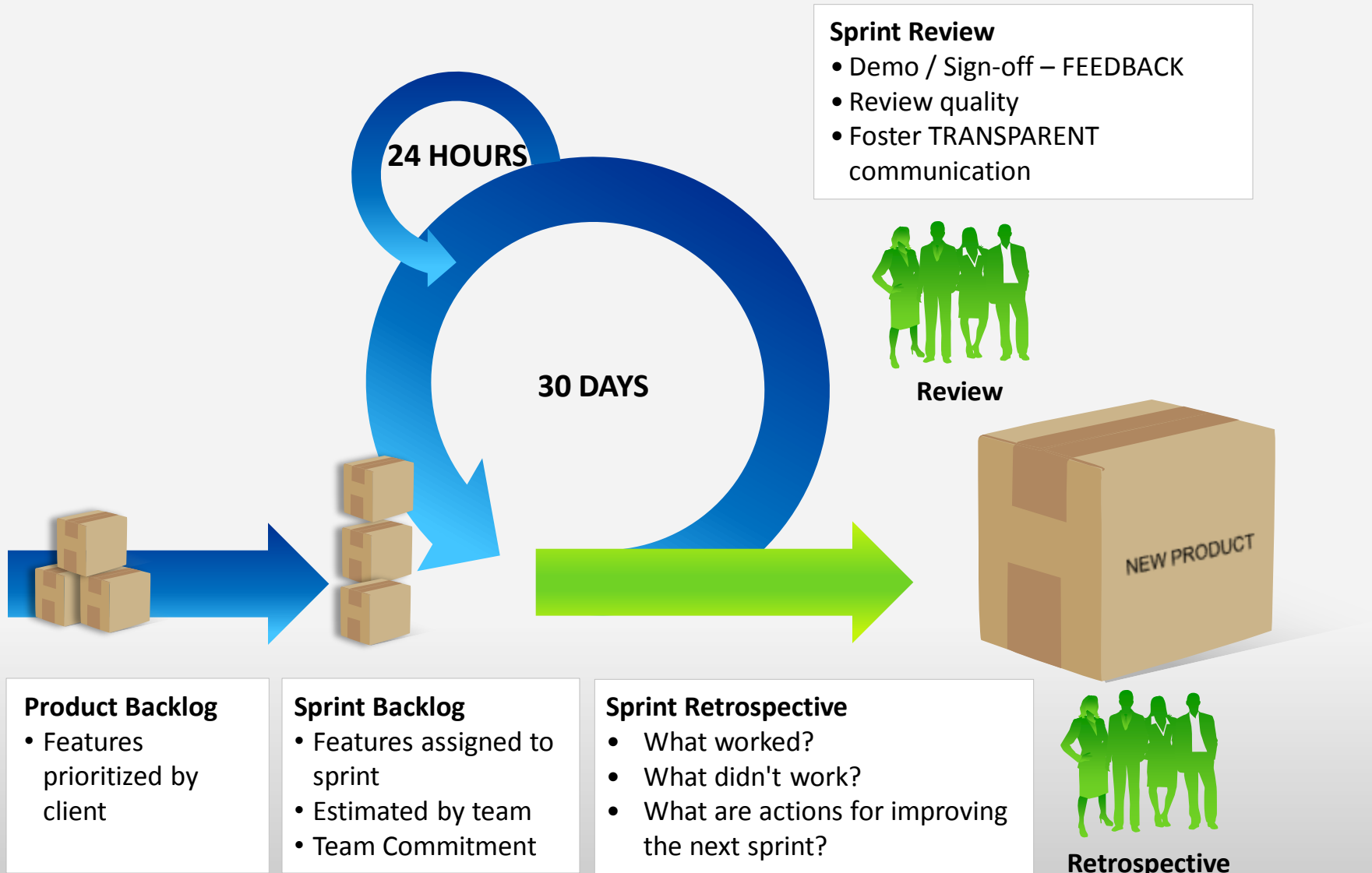
5 AGILE - Roles



AGILE Process



AGILE Process - Review & Retrospective



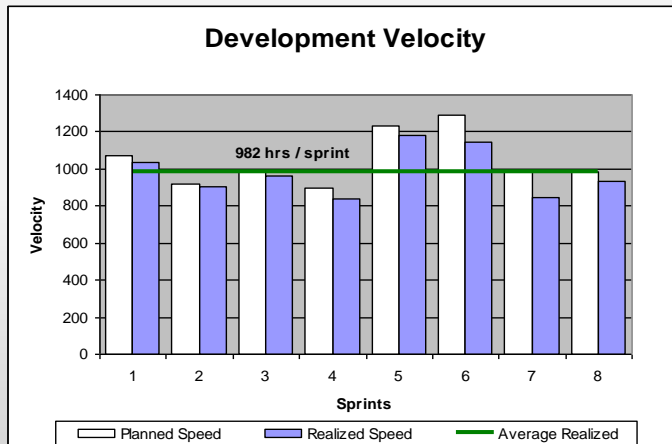
AGILE Metrics

Story ID	Story Name / Description (BOLD = Major Deliverables)	Status	Business Priority	Story Points (Ideal Days)	Sprint
1	View Trades	Planned	A	15	1
2	View Orders by Name / Account	Planned	A.1	7	1
3	View Orders by Broker	Planned	A.2	3	1
4	View Orders by Symbol	Planned	A.3	2	1
5	View Orders by Product	Planned	A.4	3	1

Product Backlog Burn-Down

- # Stories in Scope
- # Stories / Sprint
- Story Points / Sprint
- # Stories Planned / Completed

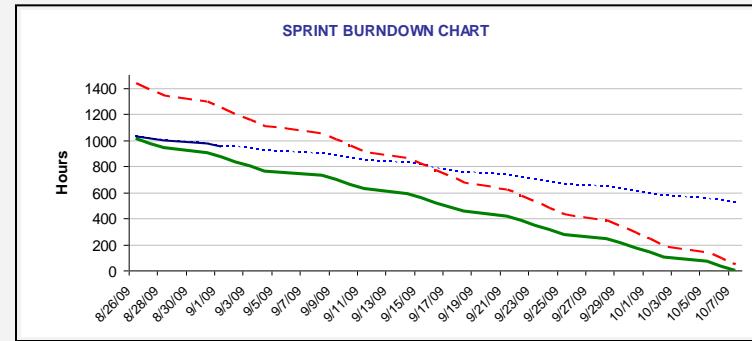
Planning



Velocity

- Ave. Velocity across Sprints
- Planned vs. Actual Velocity

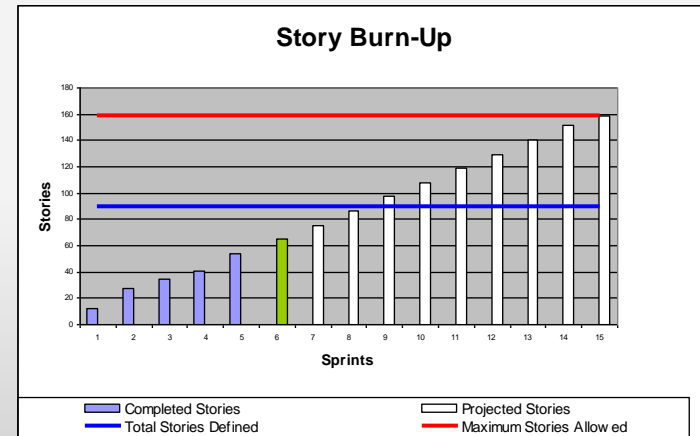
Forecasting



Sprint Burn-Down Chart

- Sprint Velocity (hrs / sprint)
- # hrs / Story
- # hrs Planned / Completed
- # hrs Leading / Lagging / Remaining

Transparency



Story Burn-Up Chart

- # Sprints Remaining
- Total Stories Defined (Scope)
- Max Stories Allowed (Budget)

Expectations

AGILE Realities

Case Studies:



- ✓ Successes
- X Challenges
- 📄 Recommendations

Case Study:



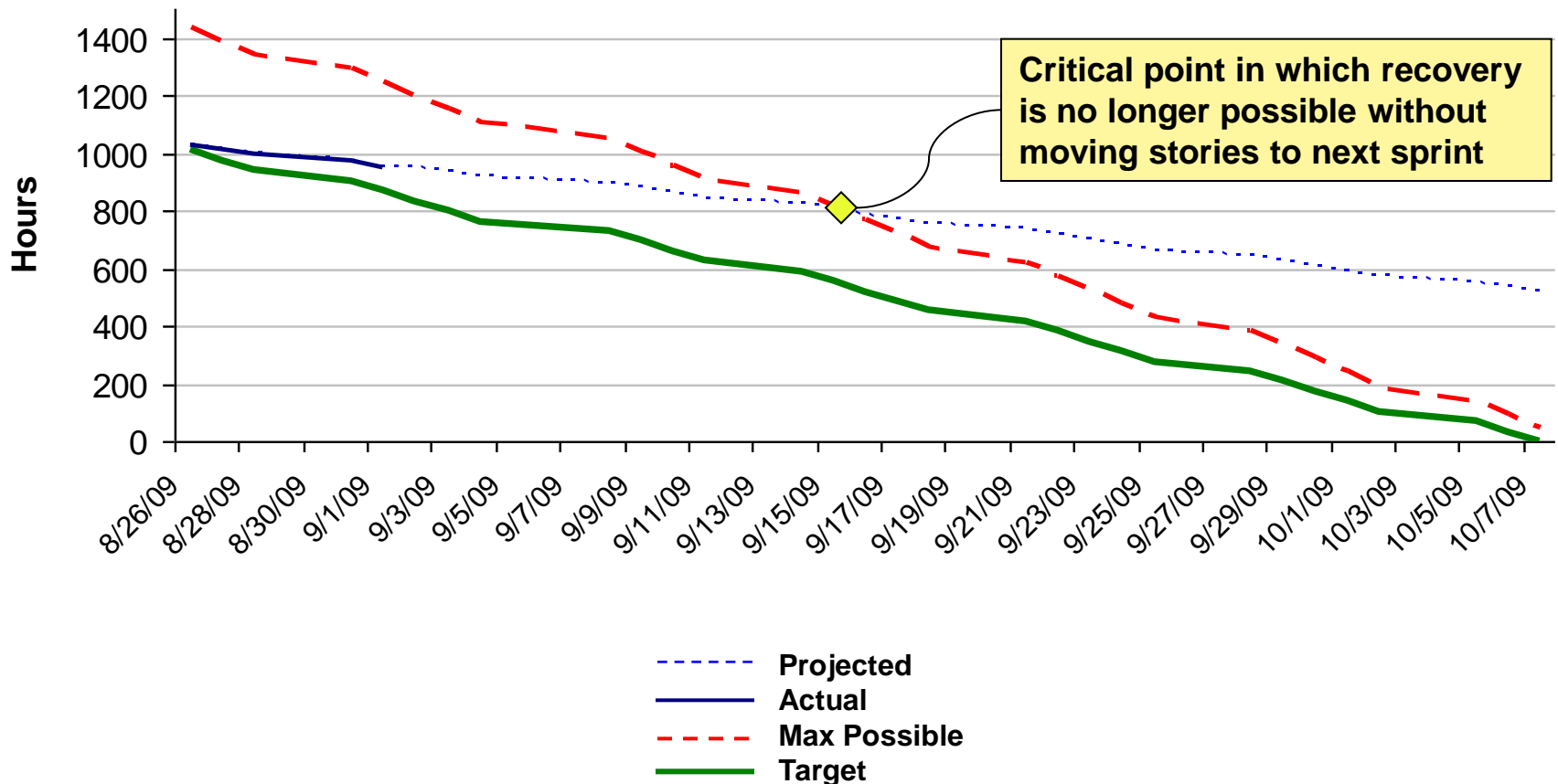
Background

- 1,200+ IT employees (infrastructure & software)
- Project types
 - Business software development for branches / clients / HQ
 - Regulatory
 - Infrastructure (facilities, hardware, network)
- Matrix organization (project staffed by functional teams)
- Highly integrated web and legacy applications
- Began deploying Scrum-based Agile in 2009; waterfall methodology deeply engrained into culture

Case Study: Edward Jones

- ✓ **Success:** Developers are focused, efficient, and self-disciplined
- 📄 **Recommendation:** Reinforce expectations that estimates are **NOT** commitments

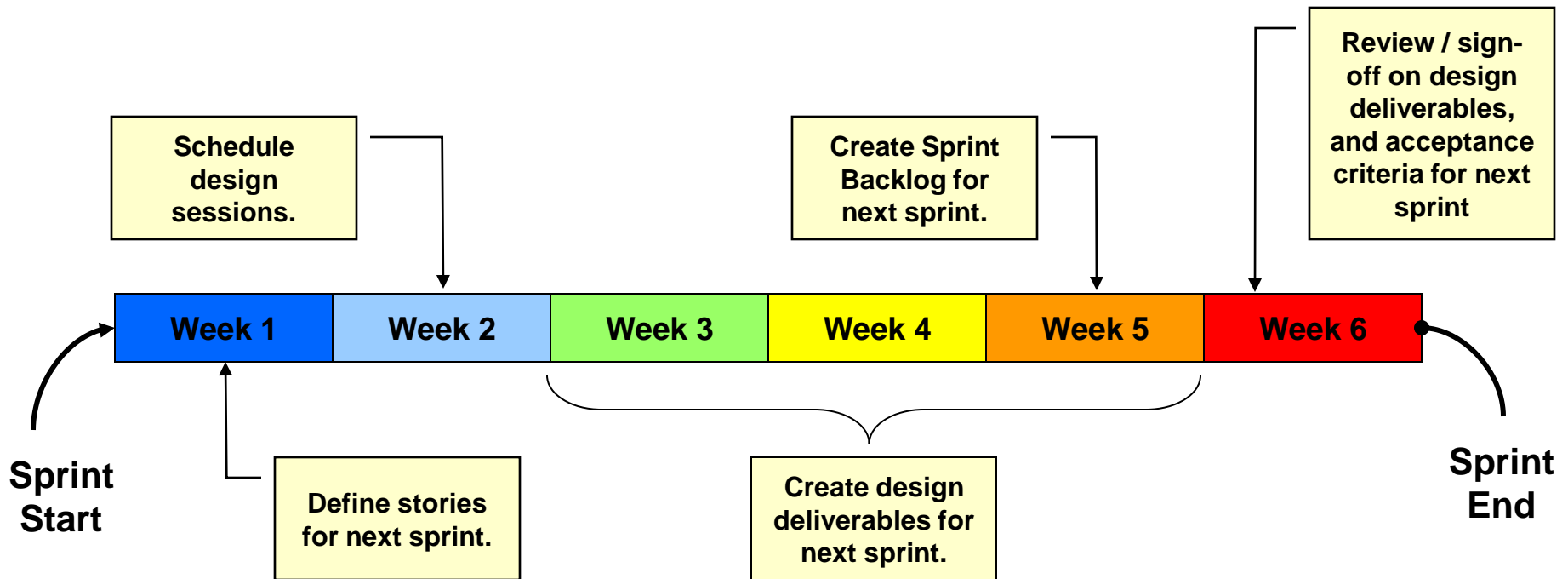
SPRINT BURNDOWN CHART



Case Study: Edward Jones

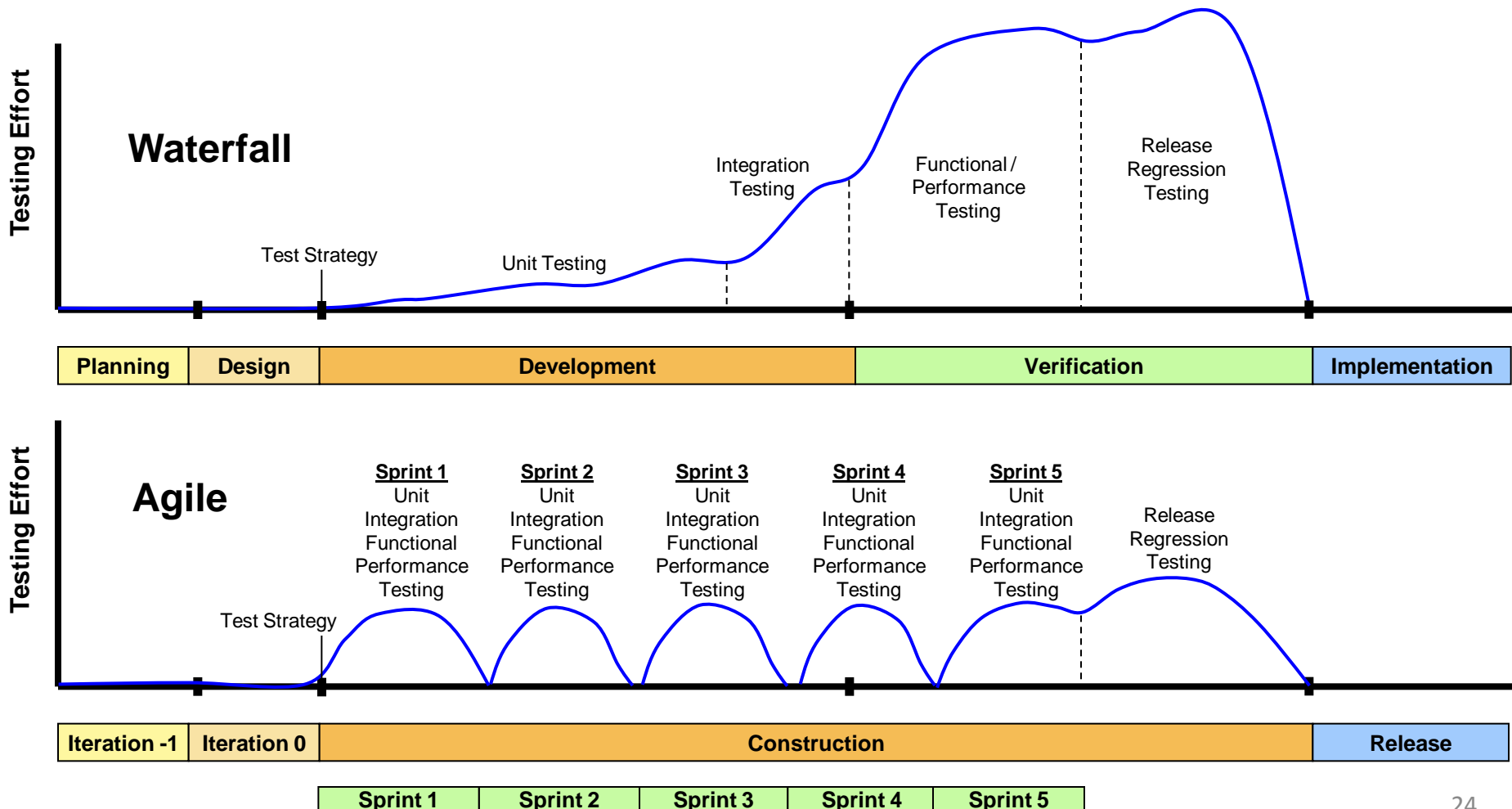
X Challenge: PL workload is a planning roller coaster.

📅 Recommendation: Plan, and stay true to, planning activities



Case Study: Edward Jones

- ✓ **Success:** Final push to implementation is less "intense."
- ✗ **Challenge:** Agile testing cycles rely on stakeholder patience.
- 📄 **Recommendation:** Set clear testing environment expectations

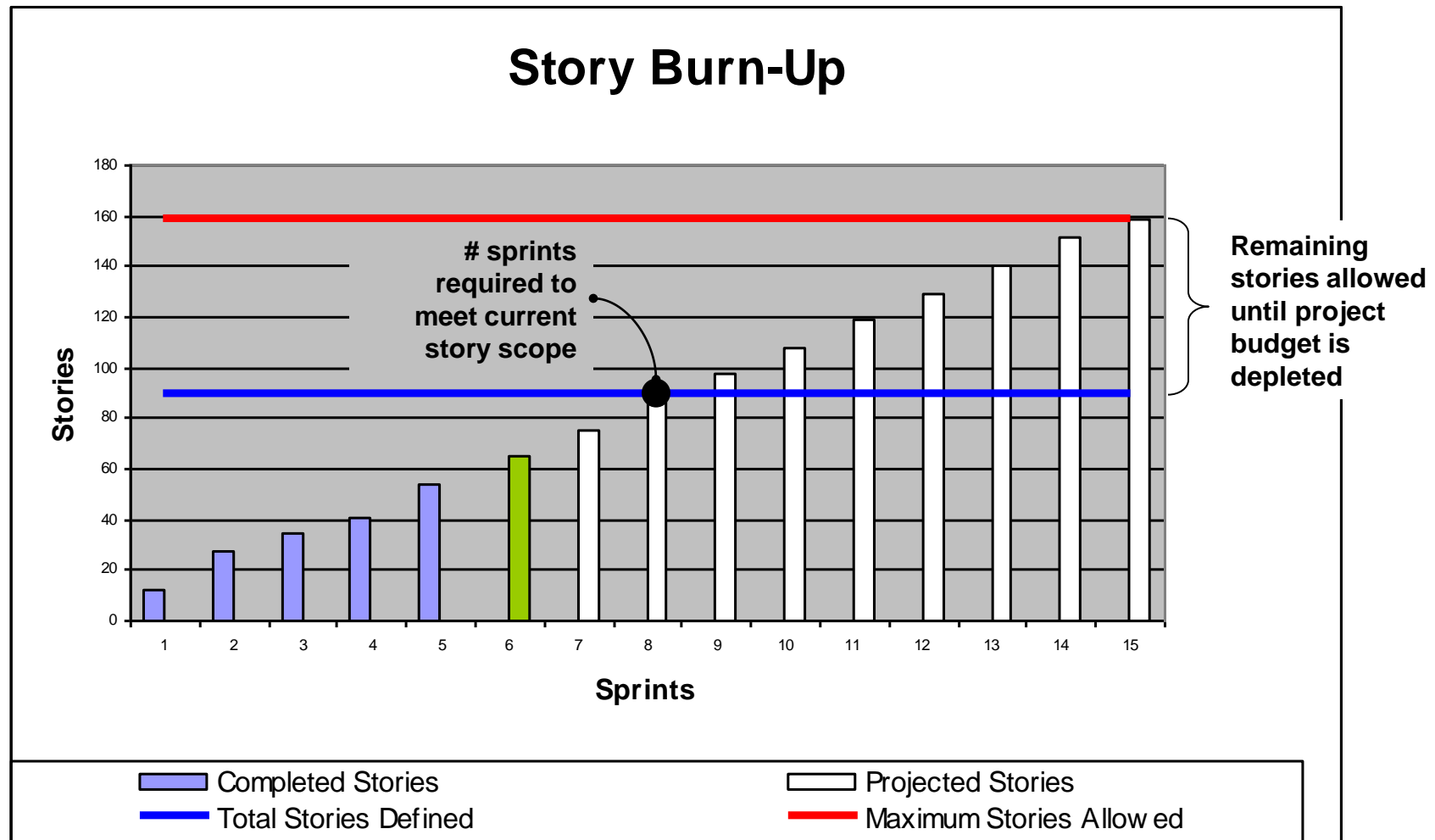


Case Study: Edward Jones

✓ **Success:** Stakeholders understand the impact of scope changes.

✗ **Challenge:** However, expectations must be set early and often.

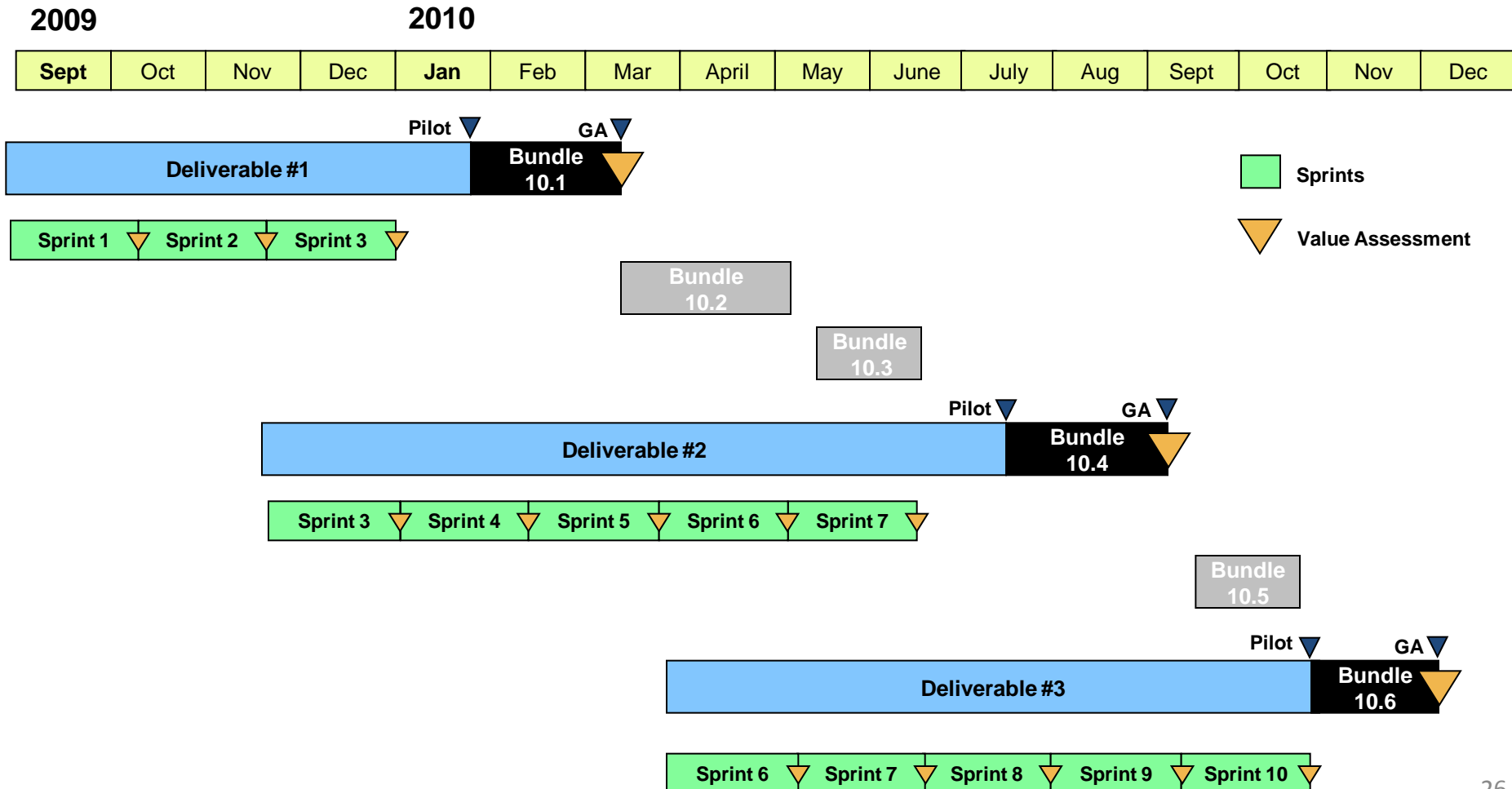
📄 **Recommendation:**



Case Study: Edward Jones

X Challenge: 30 day production deliverables often do not fit business model.

Recommendation: Clearly define a release strategy



Case Study: Edward Jones

X Challenge: Culture deeply entrenched in waterfall methodology; clear criteria needed to cauterize an AGILE mindset.

📋 Recommendation...

AGILE Eligibility Criteria

- I can commit **resources** that are focused on Agile development, and are available on-site (**100% if needed**) to communicate with sprint team members.
- I accept scope **NOT** being guaranteed, as long as I know my most important priorities are being delivered first. I realize that the approved **cost will be fixed**; when exhausted, a new phase should be submitted for approval. I realize that **lower priority** work will be "left on the table".
- I understand the importance of, and am ready to create, **test cases** to help drive out requirements, even **before any code is written**.
- I'm comfortable working **face-to-face and daily** with developers to flush out requirements, knowing that new requirements will be addressed in a next sprint.
- I understand the importance of the **Product Backlog** being the "Business" plan, and am committed to creating/updating this plan with all sprint team members on a **daily** basis.
- I understand that **vendor contracts** must align with Agile development processes.

Case Study:

Background

- 5,000+ software engineers
- Wide range of project types
 - Embedded flight software on aircraft
 - Satellite software
 - Aircraft training simulators
 - Traditional web applications
- Legacy programs with millions of lines of existing code
- Software with safety critical certification requirements
- Every program has unique constraints and complications
- Deploying a customized Scrum-based Agile process

Case Study: Boeing

✓ Successes

- Strong executive leadership support
- Dedicated group of experienced evangelist/champions
- Extensive support infrastructure
 - Training
 - Coaching
 - Outreach
- Synergistic, existing Lean initiative
- Rate of adoption by programs has exceeded expectations
- Adopting programs seeing significant improvements in productivity and quality (defect reduction)

Case Study: Boeing

X Challenges

- Every program has unique constraints.
 - Rare to have a pure, by-the-book Agile implementation
 - Agile process must be adapted to work with program constraints (we use Lean principles to inform and guide these adaptations)
 - This constrains the benefits achievable compared with a more ideal Agile team, but the benefits are still significant
- Finding cost-effective ways to train and coach multi-site, distributed teams.
- Finding effective ways to communicate our outreach message to such a large, physically dispersed enterprise.
- Implementing automated testing with legacy code bases.

Case Study: Boeing

Recommendations

- Don't buy in to the conventional wisdom of which projects are appropriate for Agile and which are not.
- Agile can provide significant benefits in most situations if you can intelligently adapt to the project's unique situation.
- The only real disqualifiers for Agile are all non-technical:
 - Someone on the team is actively opposed. Either get rid of them or don't use Agile – too easy to sabotage.
 - The customer is opposed
 - The contract does not allow any flexibility for incremental delivery
- For large organizations it is important to know and understand Lean principles.

Break-Out Session Closing

- Questions / Answers
- Thank You!
- 1-Page Take-Away Available
- "Agile Café" Roundtable (1-2:30pm)

Supporting Details

Iterative and Incremental Development

Incremental Development

*Dividing the project into various **independent** parts and developing these sub-parts at the same/different rate and integrating them when ready. Specifically, staging and scheduling in increments!*

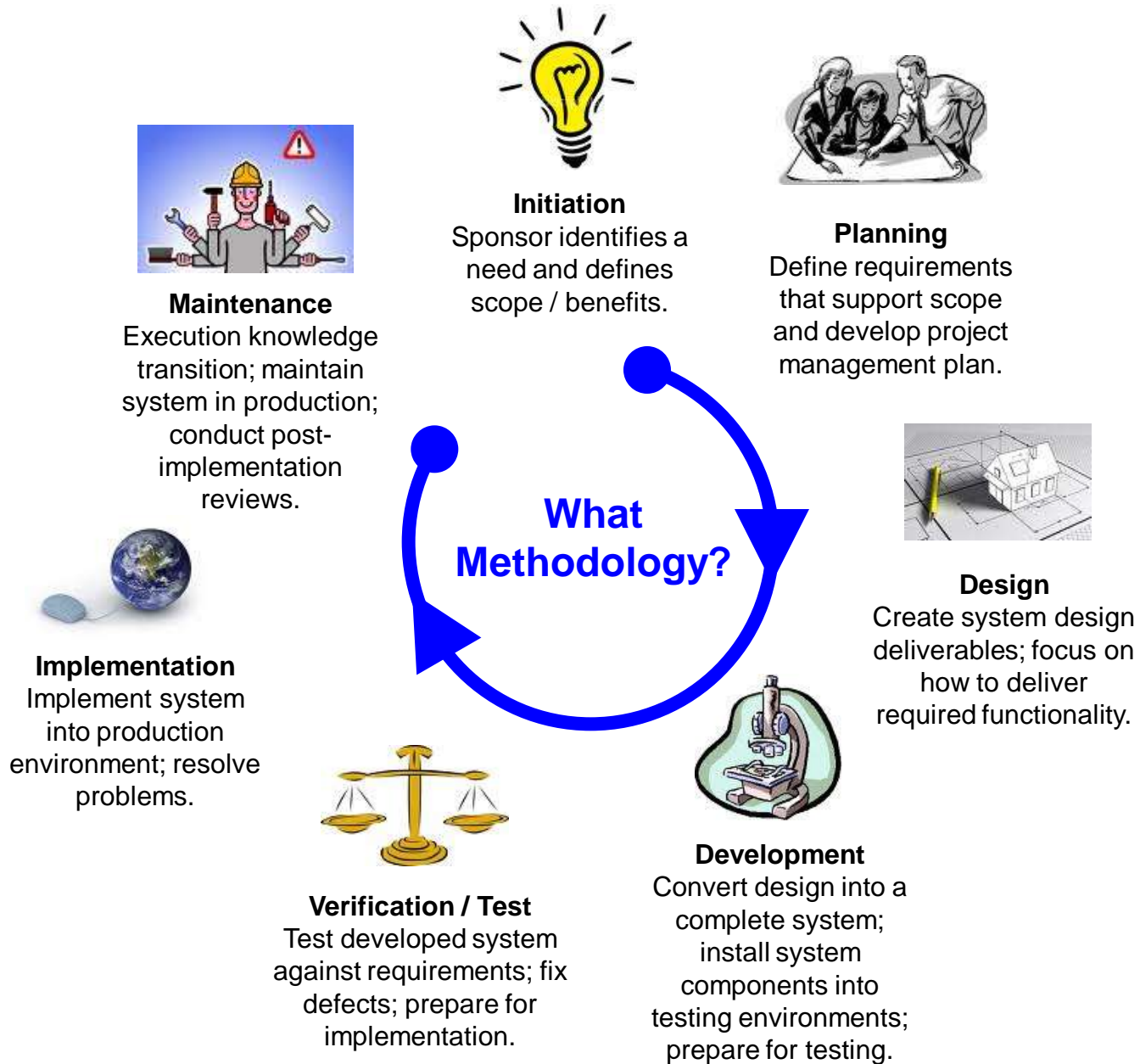
Iterative Development

Multiple iterations help “iterate” toward a final solution.

Iteration 1: Build core registration function.

Iteration 2: Enhance the core registration function.

System Development Life Cycle





Waterfall

Initiation

Sponsor identifies a need and defines scope / benefits.



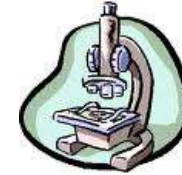
Planning

Define requirements that support scope and develop project management plan.



Design

Create system design deliverables; focus on how to deliver required functionality.



Development

Convert design into a complete system; install system components into testing environments; prepare for testing.



Verification / Test

Test developed system against requirements; fix defects; prepare for implementation.



Maintenance

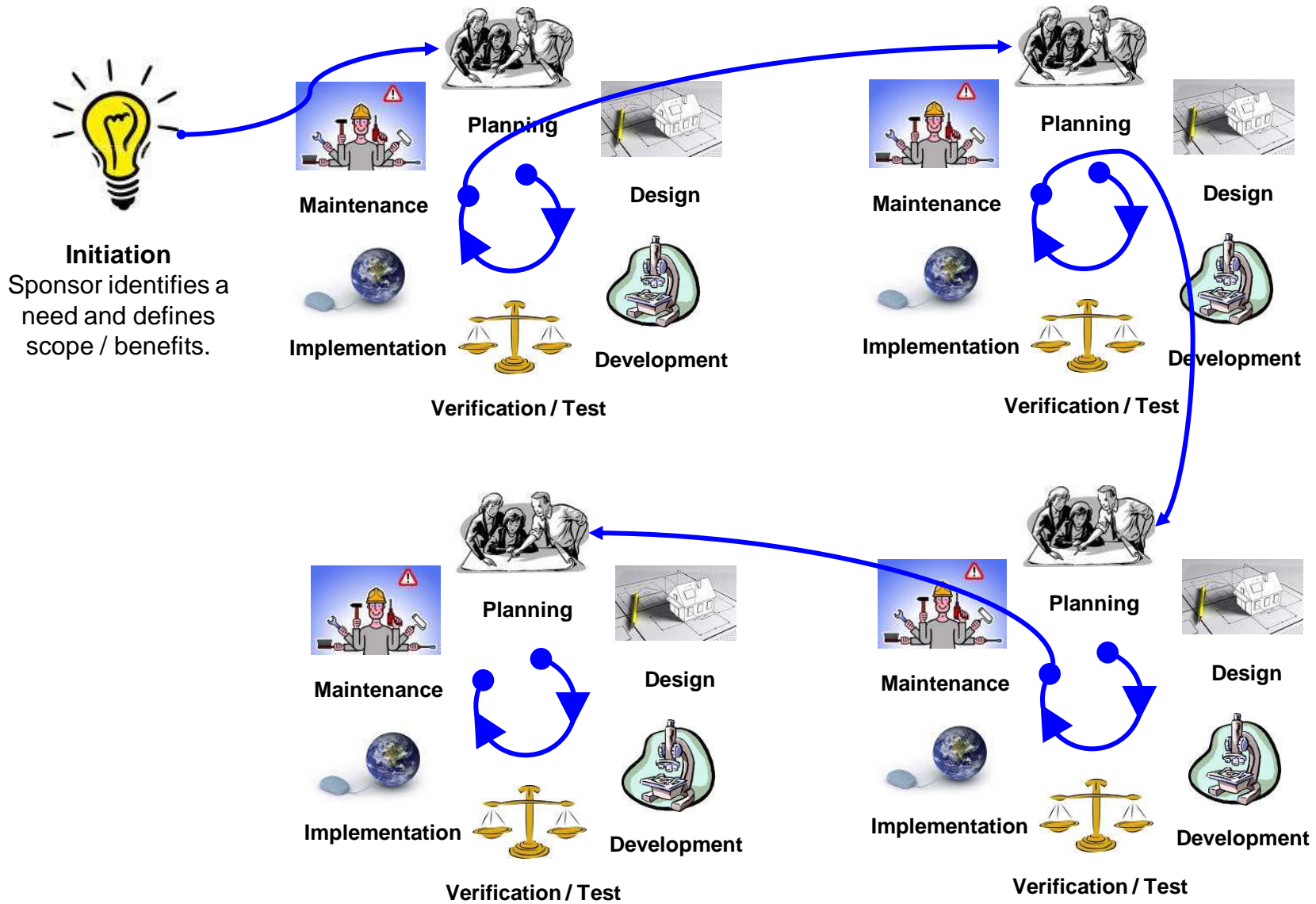
Execution knowledge transition; maintain system in production; conduct post-implementation reviews.



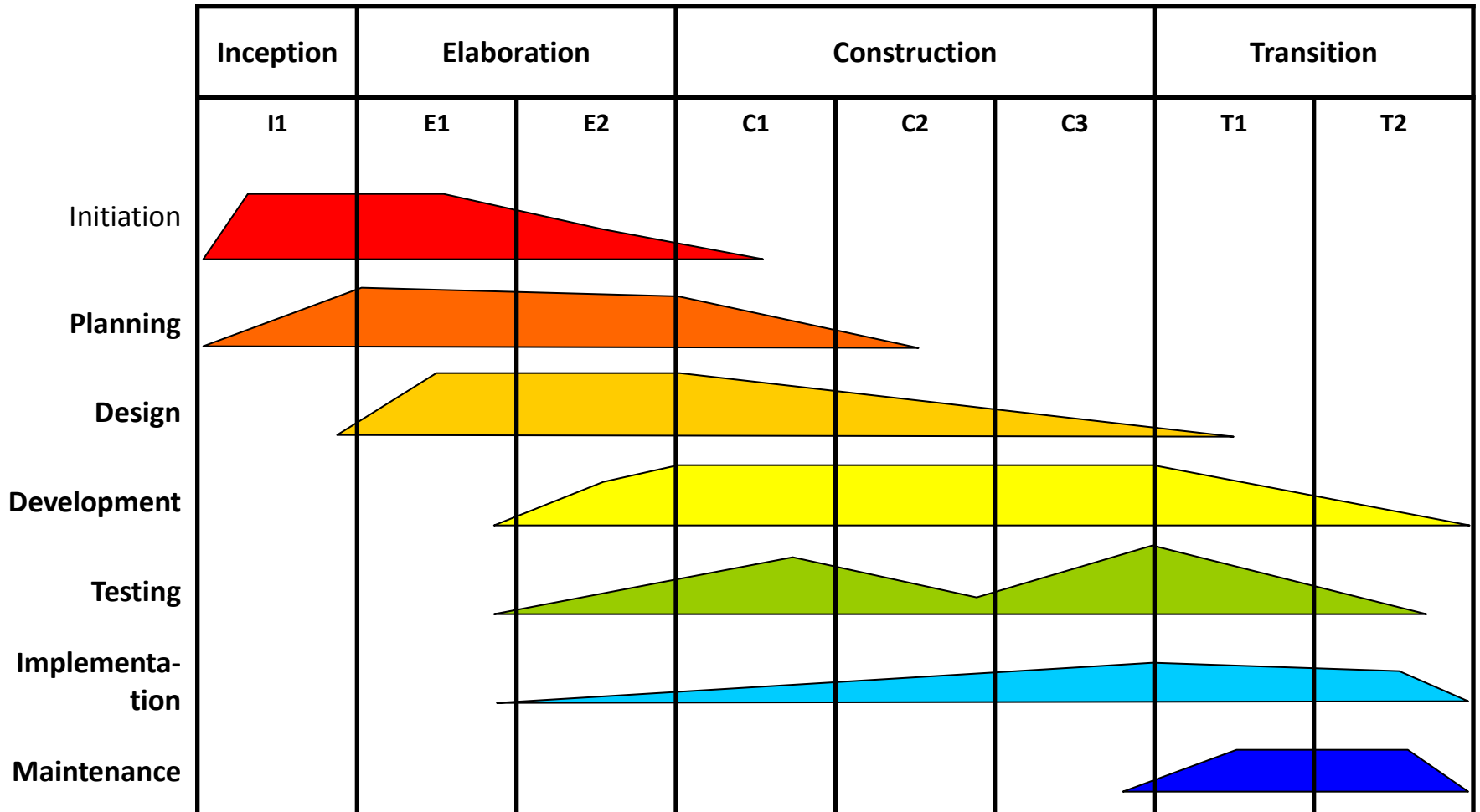
Implementation

Implement system into production environment; resolve problems.

Agile & IID



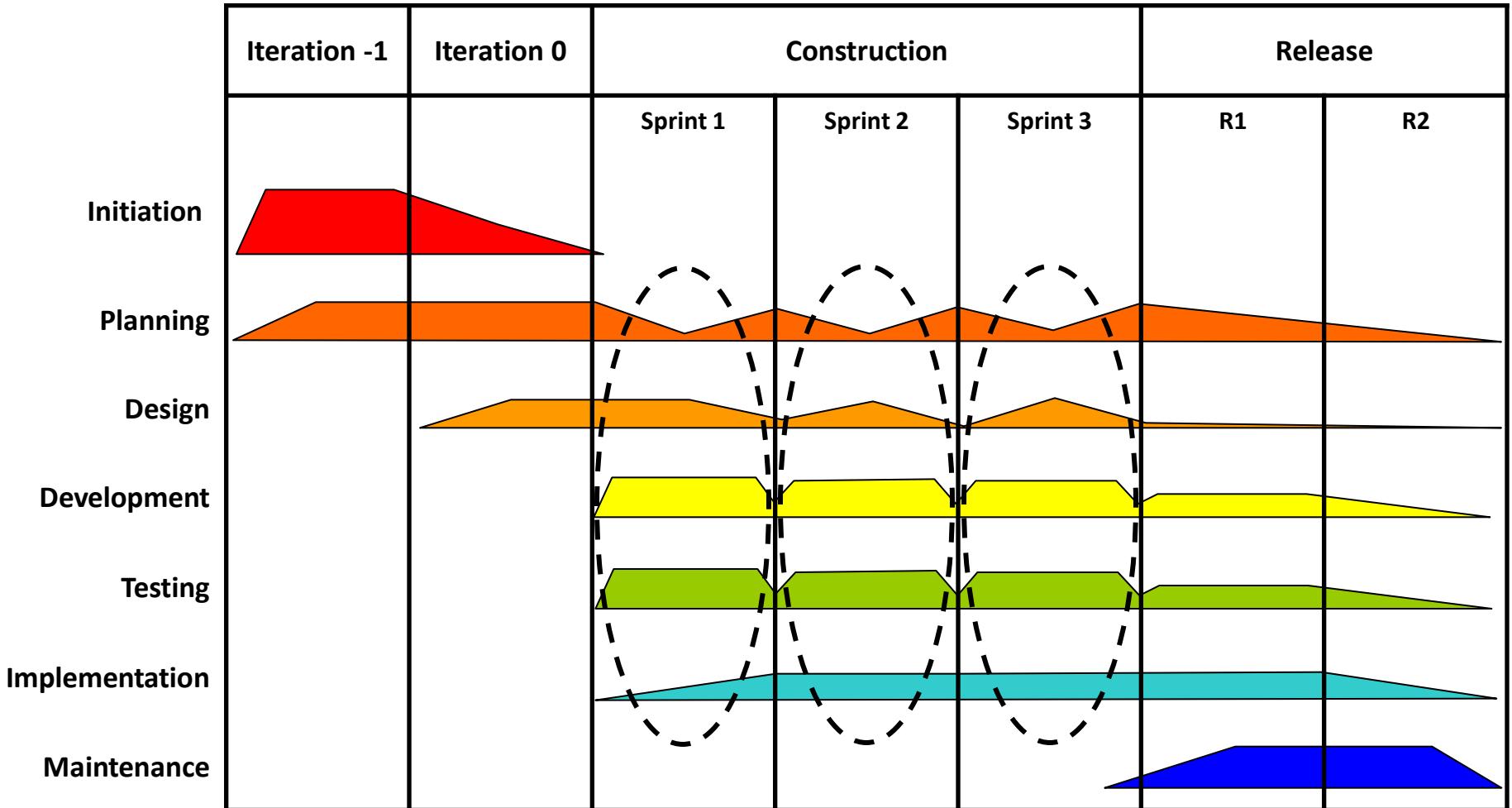
Iterative & Incremental (IID)



Time



Agile



Time