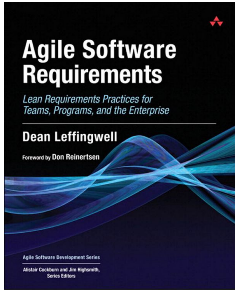
CE Department

Software Requirements Engineering

40688

These slides are designed to accompany Agile Software Requirements (2011) by Dean Leffingwell and support the university course Software Requirements Engineering, instructed by Mehran Rivadeh. Created and designed by Mahnaz Rasekhi.



Agile Software Requirements

Lean Requirements Practices for Teams, Programs, and the Enterprise

Dean Leffingwell

Foreword by Don Reinertsen

Agile Software Development Series

Alistair Cockburn and Jim Highsmith, Series Editors

Agile Software Requirements (2011)

Dean Leffingwell

The Agile Release Train

Chapter 15

Mehran Rivadeh
mrivadeh@sharif.edu
Software Requirements Engineering
October 2025 - Fall 1404 - SUT

Contents

1. Agile Release Train
2. Driving Strategic alignment
3. Institutionalizing Product Development Flow
4. Designing The Agile Release Train
5. Planning The Release
6. Tracking And Managing The Release

Introduction

Introduction

Life In The Agile Enterprise

- ➔ We see a continuous flow of releasing value to the users in small, frequent increments.
- ➔ Or a continuous build of value added to the marketplace.

The Agile Release Train

- ➔ A train that arrives on time, every time, with everyone ready and onboard.
- ➔ It's a well-organized release that people are excited about, and it runs so smoothly that it feels effortless.
- ➔ How to make each product release a successful and routine event?

Contents

1. **Agile Release Train**
2. Driving Strategic alignment
3. Institutionalizing Product Development Flow
4. Designing The Agile Release Train
5. Planning The Release
6. Tracking And Managing The Release

1. Agile Release Train

a. Purpose Of Agile Release Train

b. Principle Of The Agile Release Train

Agile Release Train

Internal Release

- It is used to evaluate the system as a whole.
- We call them PSIs, or potentially shippable increments.

External Release

- They are made generally available to our customers.
- The Release label is more appropriate.

Agile Release Train

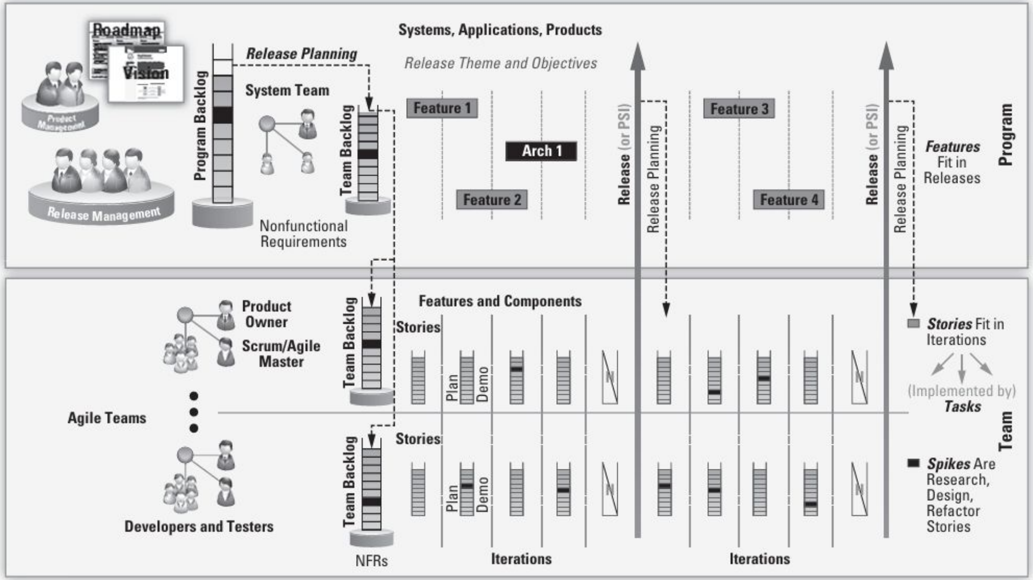
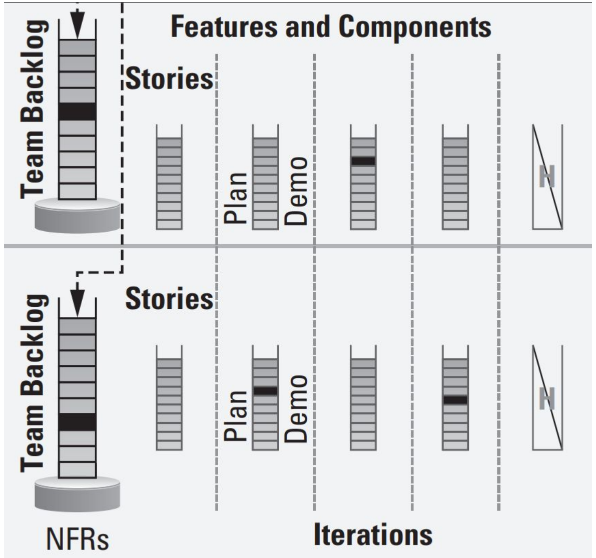


Figure 15–1 The Big Picture implying the Agile Release Train

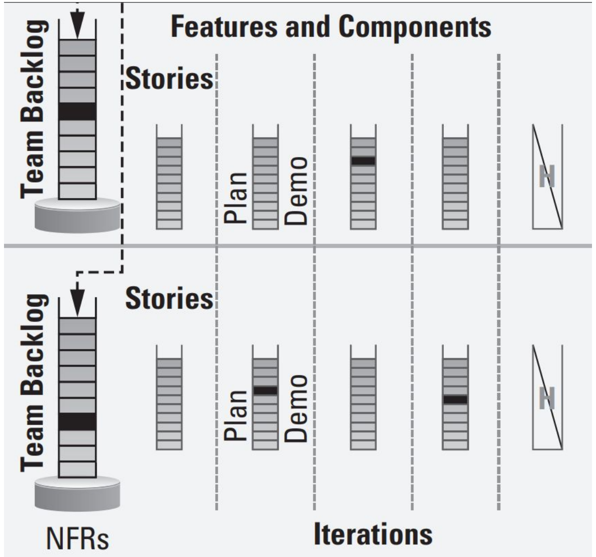
Agile Release Train

- ➔ The development of the software asset base occurs with a standard cadence of iterations that has been established by the enterprise.
- ➔ There are 4 to 5 development iterations (indicated by a full iteration backlog).
- ➔ 1 hardening iteration (indicated by an empty backlog) prior to each release increment.
- ➔ This pattern is arbitrary, and there is no fixed rule for how many times a team iterates prior to a PSI or how much, if any, time or investment in hardening is required.



Agile Release Train

- Many enterprises apply this model, creating a cadence of a shippable increment about every 90 days.
- This is a fairly natural production rhythm that corresponds to a reasonable external release frequency for customers and also provides a nice quarterly planning cadence for the enterprise.
- The length and number of iterations per release increment and the decision as to when to actually release a PSI are left to the judgment of each enterprise.
- However, the planning of an external release requires special care and attention.



Contents

1. **Agile Release Train**
2. Driving Strategic alignment
3. Institutionalizing Product Development Flow
4. Designing The Agile Release Train
5. Planning The Release
6. Tracking And Managing The Release

1. **Agile Release Train**
   a. **Purpose Of Agile Release Train**
   b. Principle Of The Agile Release Train

Purpose Of Agile Release Train

- Many of us did not start out with the ART when we headed down the agile path.
- More typically, we rolled out Scrum/XP/hybrid variants and focused on getting the teams successfully building small increments of working, tested functionality in a short timebox.
- If we were working with only one or a few teams, releasing the product did not create many additional challenges.
- We could release it when we felt like it and whenever the market required it.
- Coordinating our efforts wasn’t extraordinarily complicated either—we could do that by just talking among ourselves in the hallways, or maybe we’d have to reserve a conference room to meet with other stakeholders; quality, sales, marketing, and so on, would all be present.

Purpose Of Agile Release Train

- However,
  - as the number of teams engaged in agile development in the company increased,
  - as the enterprise grew with its successes,
  - as we acquired new teams and new products,
  - or as our customers drove us to higher levels of integration among the various components of our solution,
- a substantial problem began to emerge:

  How do we harness all that new, empowered, but potentially entropic, energy into a cohesive team of teams that can deliver ever larger and more integrated piles of value to our customers?
- To address this problem, the Agile Release Train evolved.

Contents

1. **Agile Release Train**
2. Driving Strategic alignment
3. Institutionalizing Product Development Flow
4. Designing The Agile Release Train
5. Planning The Release
6. Tracking And Managing The Release

1. Agile Release Train

a. Purpose Of Agile Release Train

b. Principle Of The Agile Release Train

Principles Of Agile Release Train

ART Advantages

- The Agile Release Train **provides alignment** and helps **manage risk** by providing Program-level cadence and synchronization.
- It is **based on agreement** and **adoption** of a **set of common** operating **principles** (OK, rules) that are followed by all teams that will be placing cargo (user value) on their particular train.

Principles Of Agile Release Train

ART Rules

- ➔ Frequent, periodic planning and release (or PSI) dates for the solution are fixed (dates are fixed, quality is fixed, scope is variable).
- ➔ Teams apply common iteration lengths.
- ➔ Intermediate, global, objective milestones are established.
- ➔ Continuous system integration is implemented at the top, system level, as well as at the feature and component levels.

Principles Of Agile Release Train

ART Rules (Cont.)

- ➔ Release increments (PSIs) are available at regular (60- to 120-day typical) intervals for customer preview, internal review, and system-level QA.
- ➔ System-level hardening iterations are used to reduce technical debt and to provide time for specialty release-level validation and testing.
- ➔ For teams to build on top of like constructs, certain infrastructure components—common interfaces, system development kits, common installs, user stores, licensing utilities, and the like—must typically track ahead.

Principles Of Agile Release Train

Although these rules may not seem that constraining, the fact is that this model requires an additional degree of agility and flexibility on all teams and stakeholders who participate.

From The Perspective Of The Team:

- To be assured of meeting a date, a team might need a primary plan and a fallback plan (or perhaps a set of options) they can deploy as necessary to make sure they can "get their cargo on the train."
- In some cases, the fallback plan can be as simple as planning to ship the old version. Even then, the team must support any new interfaces, provide backward compatibility, and be certain to not violate any other common requirements (regulatory compliance, localizations, and so on) that may be imposed on the cargo.

Principles Of Agile Release Train

From The Perspective Of Product, Program, And Executive Management:

- ➔ The plan is a result of a collaboration, which weighs the input of all stakeholders and also matches input (release requirements) to capacity (development team velocities) so that flow can be achieved.
- ➔ Rarely, if ever, do expectations of input and output match.
- ➔ Compromise is required.
- ➔ Moreover, the result of the plan is just that, a plan, and the exact scope of the final achievement can still not be known for certain up front.

Principles Of Agile Release Train

- Although implementing the Agile Release Train is a far from trivial task, implementation is a must for the agile enterprise.
- It addresses two imperatives that are necessary to achieve success within the enterprise:
    1. Driving strategic alignment across the teams
    2. Institutionalizing product development flow

Contents

1. Agile Release Train
2. **Driving Strategic alignment**
3. Institutionalizing Product Development Flow
4. Designing The Agile Release Train
5. Planning The Release
6. Tracking And Managing The Release

2. Driving Strategic Alignment

Driving Strategic Alignment

- Empowering individual agile teams to truly focus on rapid value delivery typically unlocks the raw energy, motivation, and innovation that has likely been stilted by our pre-agile process and governance models. That’s why we do it.

- However, that alone is not enough, because the teams will naturally tend toward local optimization. They’ll do what they can to deliver requirements to their customer constituency, but they have less interest (or perhaps awareness and ability) in taking a more global view. After all, having two masters is more complicated.

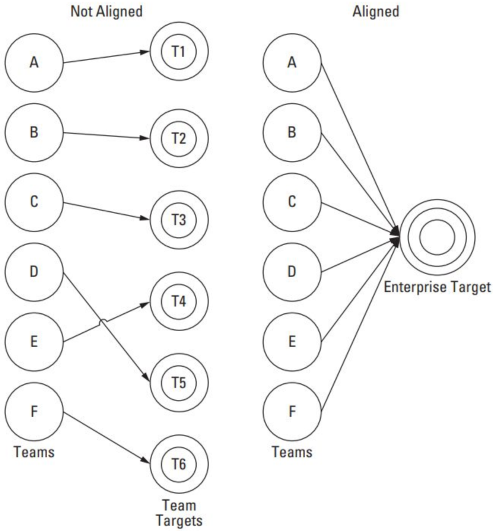
Driving Strategic Alignment

- ➔ However, in the lean enterprise, the **highest benefit** is achieved when we achieve **global optimization**.

- ➔ To do this, we must implement systems, like the ART for product programs, that **purposefully drive** the **teams toward** the **global targets**, as illustrated in Figure 15–3.

- ➔ In this way, **we can align** our **mass to** a **common direction** and achieve far more force to address the targets of opportunity.

- ➔ We can **have** both **local** and **global alignment to** a **common goal**.



Contents

1. Agile Release Train
2. Driving Strategic alignment
3. **Institutionalizing Product Development Flow**
4. Designing The Agile Release Train
5. Planning The Release
6. Tracking And Managing The Release

3. Institutionalizing Product Development Flow

Institutionalizing Product Development Flow

- In addition to driving alignment, the Agile Release Train is instrumental in institutionalizing product development flow.
- In so doing, the **ART supports** the **eight primary product development flow themes**.
- Understanding this mapping is the key to understanding the criticality and motivation for the ART itself.

Institutionalizing Product Development Flow

Theme 1: Take An Economic View

- ➔ Make trade-off decisions based on economic rationale.
    - ◆ How we frequently release the product?
- ➔ **Smaller releases** substantially **improve** the **ROI** of software development by accelerating the release of value to the customer.
- ➔ This helps **capture early market share** and drives gross margins by delivering features to the market at the time when the market values them most highly.
- ➔ In addition, we can **reprioritize features** at every planning boundary, **based on** the **current cost of delay**.

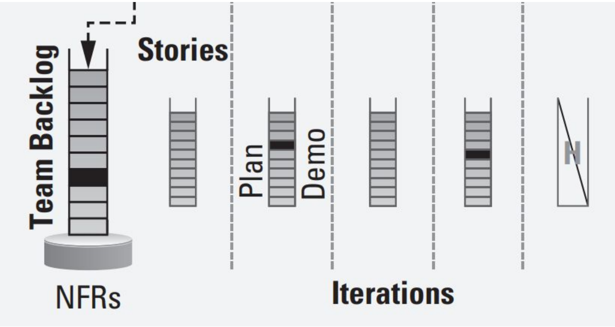
Institutionalizing Product Development Flow

Theme 2: Actively Manage Queues

- ➔ The short, frequent planning cycles of the Agile Release Train help actively manage queue lengths across the enterprise.

Team backlogs

- ➔ These queues of waiting stories are generally limited to about the amount of work that can be accomplished in a single PSI.
- ➔ Planning much beyond that is generally not very productive for the teams, because strategic priorities could change at any release boundary.

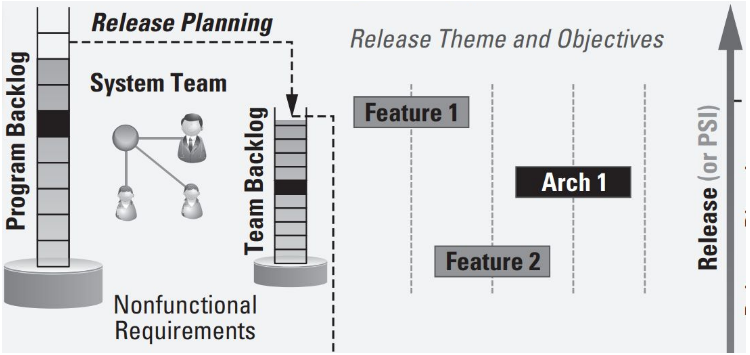


Institutionalizing Product Development Flow

Theme 2: Actively Manage Queues

Release (Program) Backlogs

- ➔ These queues of waiting features are typically limited to those features that can realistically be implemented in the next release or two.
- ➔ Beyond that, product managers understand that they may be overinvesting in elaboration of features that will never see the light of day.

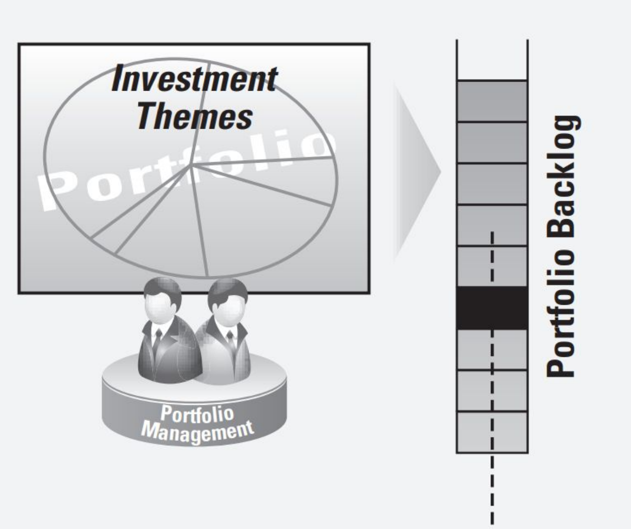


Institutionalizing Product Development Flow

Theme 2: Actively Manage Queues

Portfolio Backlogs

- ➔ These queues of waiting epics and future projects are typically limited to those epics that could likely find their way to release planning in the next six months or so.
- ➔ Too early, or too in-depth, investment in business cases for projects that will not be implemented is a form of waste.



Institutionalizing Product Development Flow

Theme 3: Understand And Exploit Variability

- ➔ Since a high degree of variability is inherent in software development, frequent, cadence-based re-planning provides the opportunity to adjust and adapt to circumstances as fact patterns change.
- ➔ New, unanticipated opportunities can be exploited by quickly adapting plans.
- ➔ Critical paths and bottlenecks become clear.
- ➔ Resources can be adjusted to optimize throughput and better avoid unanticipated delays.

Institutionalizing Product Development Flow

Theme 4: Reduce Batch Sizes

- Large batch sizes create unnecessary variability and cause delays in delivery and quality.
- ART reduces batch sizes by releasing to development only those features that:
  - are prioritized,
  - are elaborated sufficiently for development,
  - and are sized to fit within the next release cycle.
- This helps avoids overloading the development teams with multiple development projects, which otherwise causes multiplexing, thrashing, and loss of productivity.
- Face-to-face planning provides high-bandwidth communication and instant feedback, so the transport (handoff) batch delay between teams is minimized.

Institutionalizing Product Development Flow

Theme 5: Apply WIP Constraints

- ➔ Teams plan their own work and take on only the amount of features that their velocity indicates they can achieve.
- ➔ This forces the input rate (agreed-to, negotiated release objectives) to match capacity (what the teams can do in the release - **W**ork **I**n **P**rocess).
- ➔ The current release timebox prevents uncontrolled expansion of work so that the current release does not become a “feature magnet” for new ideas.
- ➔ The global WIP pool, consisting of features and epics in the enterprise backlog, is constrained by the local WIP pools, which reflects the team’s current backlog as driven by the current PSI.
- ➔ Limiting WIP increases response time to new, higher-priority activities.

Institutionalizing Product Development Flow

Theme 6: Control Flow Under Uncertainty → Cadence And Synchronization

- ➔ Cadence and synchronization help us manage uncertainty and variability by keeping accumulated variances to single interval.
- ➔ In the ART, we achieve this through periodic planning (cadence) and integrating (synchronization).

Institutionalizing Product Development Flow

Theme 6: Control Flow Under Uncertainty → Cadence And Synchronization

Planning

- ➔ The release train planning cadence makes planning predictable and lowers transaction costs (facilities, overhead, travel).
- ➔ Planning can be scheduled well in advance, allowing participation by all key stakeholders in most planning events and making face-to-face information transport reliable, efficient, and predictable.
- ➔ Periodic re-planning (resynchronization) allows us to limit variance and misalignment to a single planning interval.

Institutionalizing Product Development Flow

Theme 6: Control Flow Under Uncertainty → Cadence And Synchronization

Integrating

- The regular, system-wide integration provides high-fidelity system tests and objective assessment of project status at regular intervals.
- Transaction costs are lowered as teams prioritize investment in the infrastructure necessary for continuous integration, automated testing, and more automated deployment.
- Since planning is bottom-up, (performed by the teams and based on team's actual known velocity) and short-term, delivery becomes predictable.
- Most all that has been planned should be reliably available as scheduled.

Institutionalizing Product Development Flow

Theme 7: Get Feedback As Fast As Possible

- ➔ The fast feedback of the iteration and release cycle allows us to take fast corrective action.
- ➔ Even within the course of a PSI, feedback is no more than two weeks (or the iteration length) away.
- ➔ Small incremental releases to customers allow us to track more quickly to their actual, rather than anticipated, needs.
- ➔ Incorrect paths can be abandoned more quickly (at worst, at the next planning cycle).

Institutionalizing Product Development Flow

Theme 8: Decentralized Control

- ➔ Release plans are prepared by the teams that are doing the actual implementation, rather than by a planning office or project management function.
- ➔ Commitments to the plans are bottom-up based on each individual's commitment to teammates and team-to-team commitments reached during the planning cycle.
- ➔ Once planned, the teams are responsible for execution, albeit subject to appropriate lightweight governance and release management.
- ➔ Agile project management tooling automates routine reporting; management does not have to slow down and annoy the teams to assess actual status.

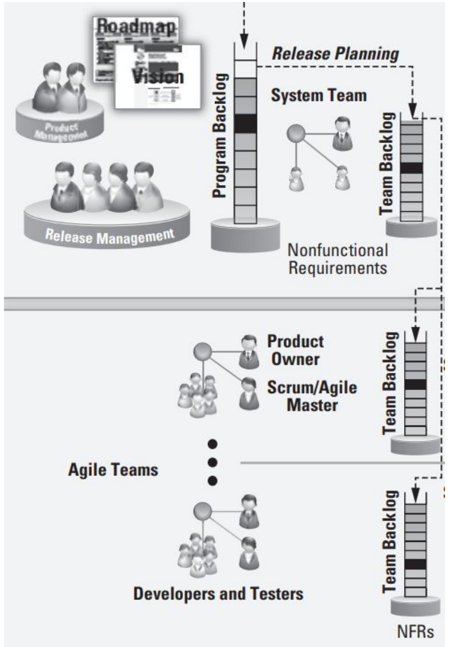
Contents

1. Agile Release Train
2. Driving Strategic alignment
3. Institutionalizing Product Development Flow
4. **Designing The Agile Release Train**
5. Planning The Release
6. Tracking And Managing The Release

4. Designing The Agile Release Train

Designing The Agile Release Train

- One initial activity is to determine the release train domain:
  - Who will be planning and working together.
  - What products, services, features, or components the train will deliver.
- In the Big Picture, we've indicated that there is some collection (or pod) of agile teams that constitute a program.



Designing The Agile Release Train

- ➔ That is often the case, and in the smaller enterprise or business unit, the ART domain consists of everyone on the team who will participate in the outcome.
- ➔ If the assets you are building can be built with five to eight agile teams, then the planning domain is the program, and not much more thought is required.
- ➔ However, in the larger enterprise, there may be dozens (or more) of such teams, and planning everything together is not feasible.
- ➔ In that case, we must first determine who will be on the train.
- ➔ Considerations should include the following.

Designing The Agile Release Train

- Trains should be focused on a single, primary product, solution, or value theme objective.
- Trains work best when between 50 to 100 people, including stakeholders outside the team, contribute to the train.
- Teams with features and components that have a high degree of interdependencies should plan and work together.
- Locale is a major consideration. Wherever possible, train teams should be co-located, or at least geographic distribution should be as limited as feasible, because that simplifies planning logistics and cooperation among the teams.

Contents

1. Agile Release Train
2. Driving Strategic alignment
3. Institutionalizing Product Development Flow
4. Designing The Agile Release Train
5. **Planning The Release**
6. Tracking And Managing The Release

5. Planning The Release

Planning The Agile Release Train

- Once the parameters and the cadence for the ART have been established, the teams can establish a release-planning schedule for the train.

- Since the dates for the PSIs are fixed, the release planning dates can be fixed as much as a year in advance.

- This helps lower facility, travel, overhead, and other transaction costs associated with the event.

- Given the importance of the event in driving strategic alignment, planning and executing the release event is a project unto itself.

- We’ll cover release planning thoroughly in the chapter 16.

Planning The Agile Release Train

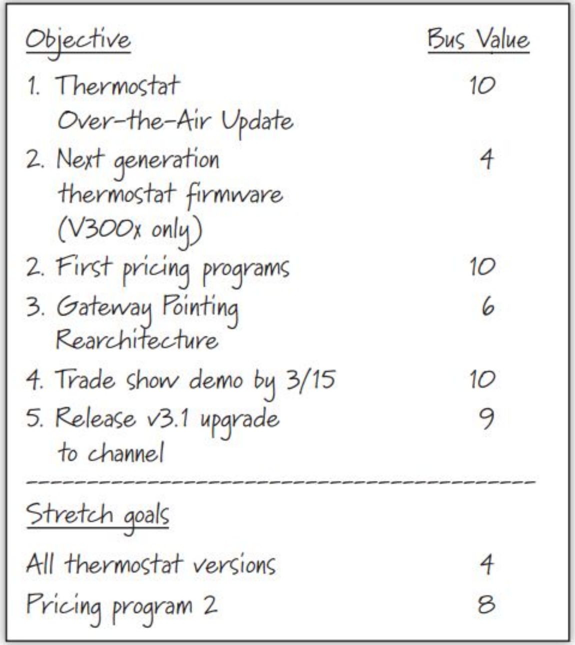
Release Objectives

- One important result of the release planning process is a set of release objectives, which define the individual team and aggregate goals of the release.
- These quantitative objectives are a key artifact of the release planning session and provide us with an important baseline for release governance and tracking.

Planning The Agile Release Train

Release Objectives

- ➔ Each objective will have been ranked by business value, as the example in the Figure illustrates.
- ➔ Most of the objectives will be features from the backlog, and this gives us the targets we need to track and manage the release.
- ➔ The figure is an example of release objective, ranked by business value



Objective	Bus Value
1. Thermostat Over-the-Air Update	10
2. Next generation thermostat firmware (V300x only)	4
2. First pricing programs	10
3. Gateway Pointing Rearchitecture	6
4. Trade show demo by 3/15	10
5. Release v3.1 upgrade to channel	9
Stretch goals	
All thermostat versions	4
Pricing program 2	8

Contents

1. Agile Release Train
2. Driving Strategic alignment
3. Institutionalizing Product Development Flow
4. Designing The Agile Release Train
5. Planning The Release
6. **Tracking And Managing The Release**

6. Tracking And Managing The Release

Tracking And Managing The Release

- With the quality and date fixed, it is certain that some amount of adjustment to content will be needed during the course of the PSI.
- In support of this need, the enterprise will likely have implemented some form of **agile requirements** or **project management tooling**, which provides support for the higher-level status views needed by product managers and other stakeholders.
- Such tooling should **provide hierarchical**, **release-level burndown** so that the program can assess, on an aggregate basis, exactly where they stand within the release, as Figure 15–5 shows.

Tracking And Managing The Release

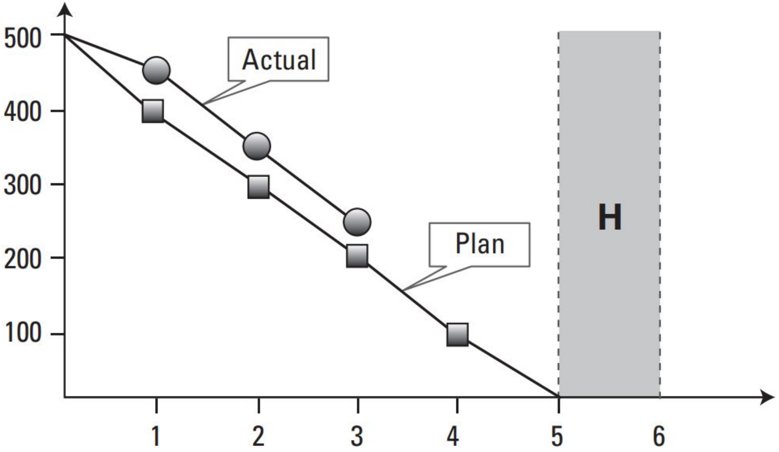


Figure 15–5 Release-level burndown

Tracking And Managing The Release

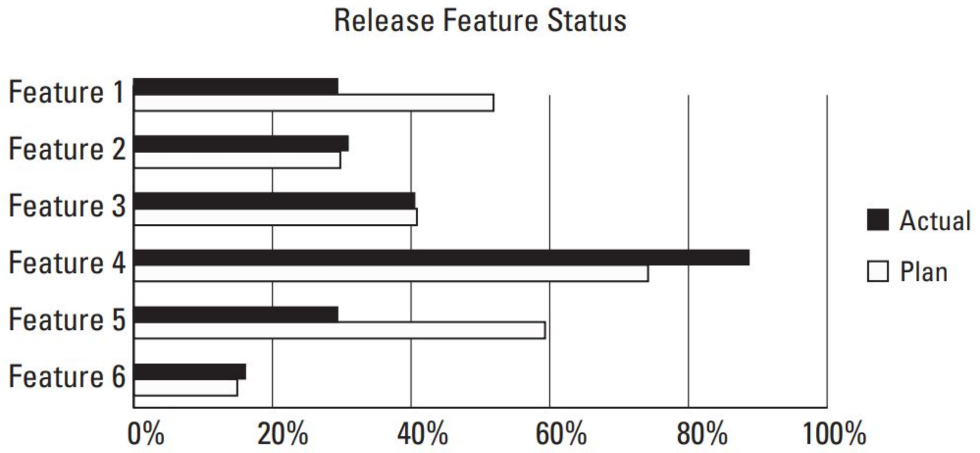


Figure 15–6 Release status by feature

End of Chapter 15

Contributions

- Author of Reference Book: **Dean Leffingwell**
- Course Instructor: **Mehran Rivadeh**
- Slide Creator: **Mahnaz Rasekhi**
  - These slides are primarily based on Agile Software Requirements by Dean Leffingwell, with occasional adaptations to enhance clarity and engagement.