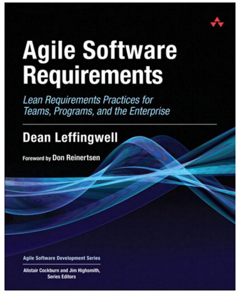
CE Department

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

Agile Requirements For The Portfolio

Chapter 5

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

Contents

1. Introduction To The Portfolio Level

Introduction To The Portfolio Level

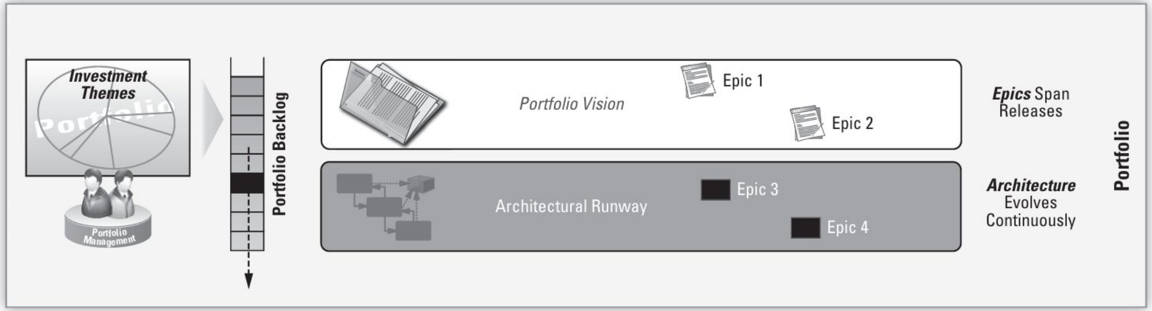
- For many software enterprises the team model plus the program model may be all that the teams need to manage system requirements in an agile manner.
  - **What Software Enterprises?**
    - Modest scope of 100 or so practitioners.
    - Develop and manage only one or two products.
  - **The Team Model:** user stories, tasks, and acceptance tests
  - **The Program Model:** features and nonfunctional requirements
- In this context, driving releases with a feature-based vision and driving iterations with stories created by the teams may be all that is required.

Introduction To The Portfolio Level

There is another class of enterprises.

- ➔ Those employ hundreds to thousands of practitioners.
- ➔ Those that have many products.

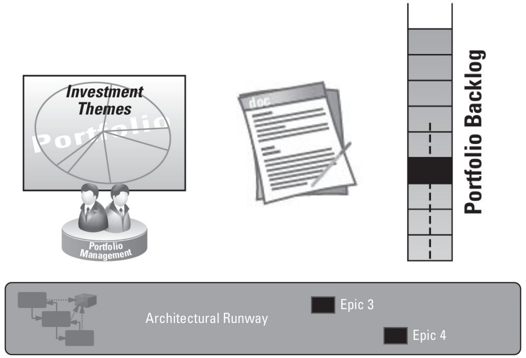
Wherein the governance and management model for new software asset development needs additional artifacts and still higher levels of abstraction.



Introduction To The Portfolio Level

The Portfolio level introduces:

- ➔ Two new artifact types
  - ◆ Investment Themes
  - ◆ Epics
- ➔ A new backlog → The portfolio backlog
- ➔ A new team → The portfolio management team
- ➔ Portfolio vision
- ➔ Architectural runway



Contents

2. Investment Themes

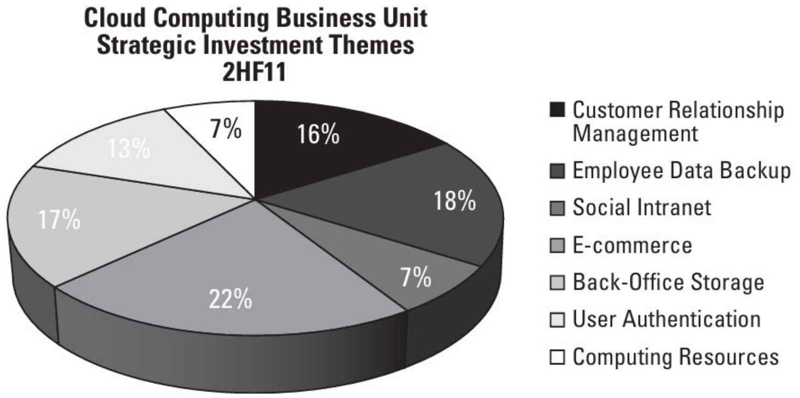
Investment Themes

That’s where everything starts.

- ➔ Investment themes (or product themes) represent the set of initiatives that drive the enterprise’s investment in systems, products, applications, and services.
- ➔ Investment themes represent key product or service value propositions that provide marketplace differentiation and competitive advantage.

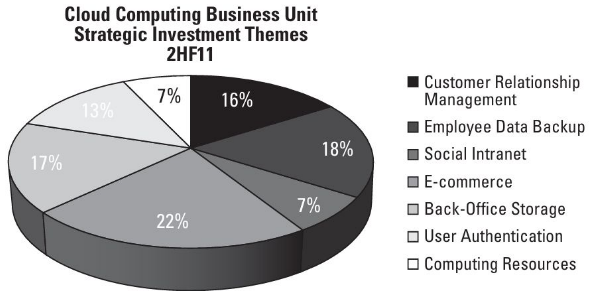
Investment Themes

- The set of strategic investment themes for an enterprise, or business unit within an enterprise, establishes the relative investment objectives for the entity.
- Each “partition” in the graphic represents a specific development initiative and an associated budget.



Investment Themes

- Within the partition (budget allocation), managers are empowered to develop the initiative in whatever way makes the most economic and business sense for the enterprise.
- However, they generally may not exceed the budget or borrow resources from other themes without agreement with those stakeholders.
- With this process, the enterprise exercises its fiduciary responsibility by driving investment to the agreed-to business priorities.

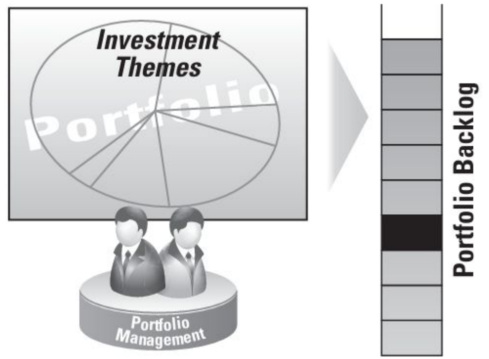


Contents

3. Portfolio Management Team

Portfolio Management Team

- ➔ The derivation of these decisions is the responsibility of the portfolio management function.
- ➔ Those individuals who have ultimate responsibility for the individual lines of business.
- ➔ In larger enterprises, this typically happens at the business unit level based on an annual or twice-annual budgeting process.



Portfolio Management Team

The portfolio management team **makes its decisions based on** some combination of the following:

1. Investment in **existing** product offerings
   a. enhancements, support, and maintenance
2. Investment in **new** products and services
   a. products that will enhance revenue
   b. and/or gain new market share in the current or near-term budget period

Portfolio Management Team

The portfolio management team **makes its decisions based on** some combination of the following:

3. Investment in **futures**
    a. advanced product and service offerings that require investment today but will not contribute to revenue until outlying years.
4. **Reducing** investment (sunset strategy)
    a. for existing offers that are nearing the end of their useful life.

Portfolio Management Team

- ➔ Themes have a much longer life span than epics.
- ➔ A set of investment themes may be largely unchanged for up to a year or more.
- ➔ To provide ongoing governance and visibility into the investments, the portfolio management team may be assisted by a project management office (PMO).

Contents

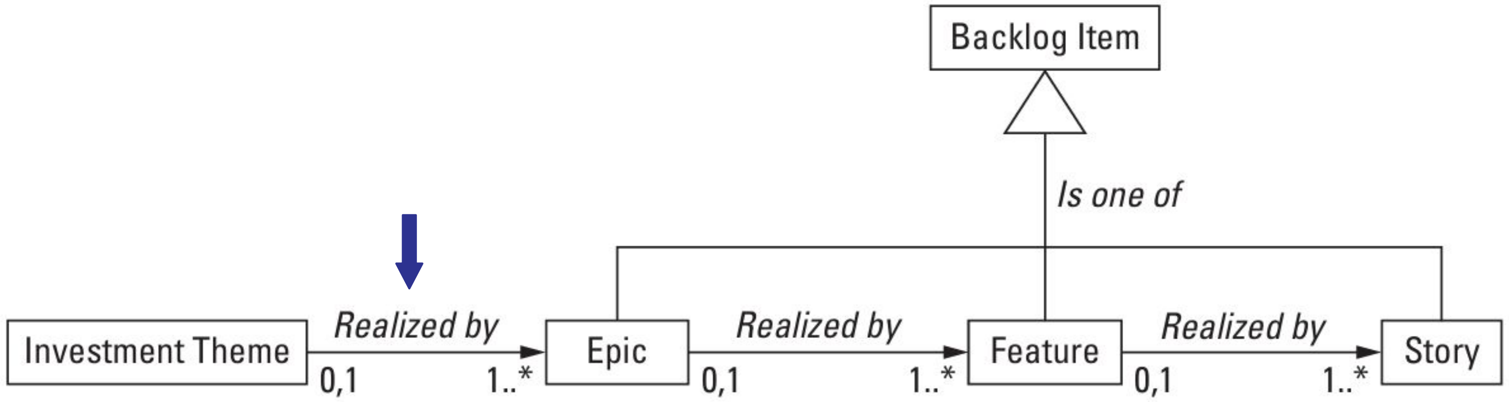
4. Epics And The Portfolio Backlog

Epics

- ➔ The set of strategic investment themes drive all new development, and requirements epics are derived from these decisions.
- ➔ Epics are large-scale development initiatives that realize the value of investment themes.
- ➔ Epics are the highest-level requirements artifact that we will use to coordinate development.
- ➔ In the requirements model, they sit between investment themes and features.

Epics

- Epics are typically driven (parented by) investment themes.
  - But some epics can be independent (they do not require a parent in order to exist).

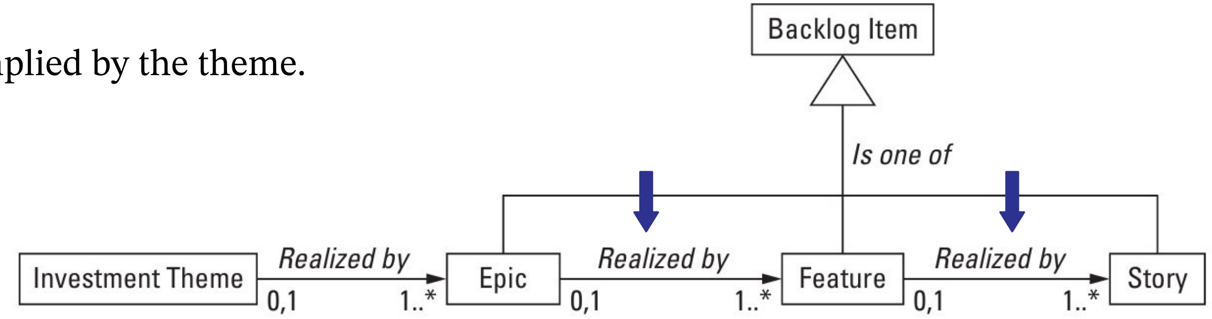


Epics

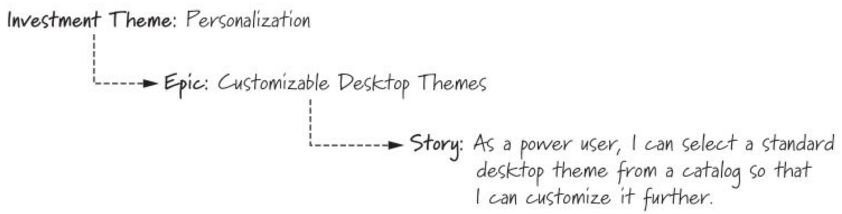
- ➔ Epics are not implemented directly.

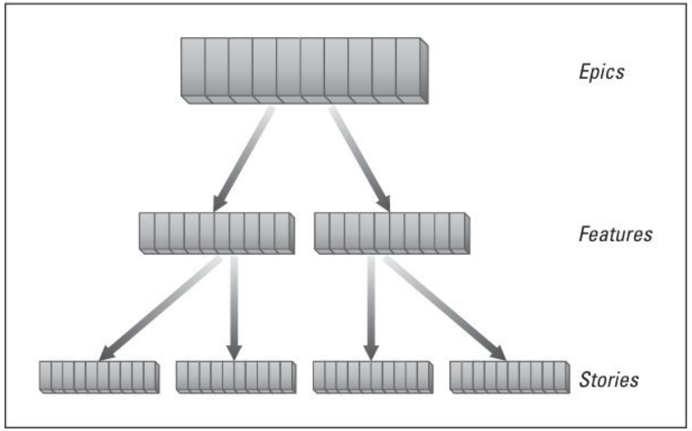
  Instead, they are broken into features, which, in turn, are broken into user stories, which are the primitives used by the teams for actual coding and testing.
- ➔ Epics are not directly testable.

  Instead, they are tested by the acceptance tests associated with the features and stories that implement them.
- ➔ Epics deliver the value implied by the theme.



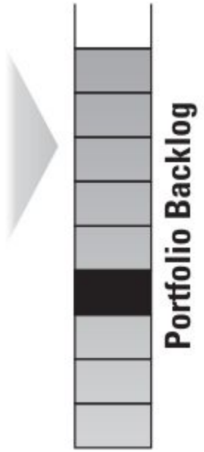
Epics


Investment Theme: Personalization
Epic: Customizable Desktop Themes
Story: As a power user, I can select a standard desktop theme from a catalog so that I can customize it further.


Epics
Features
Stories

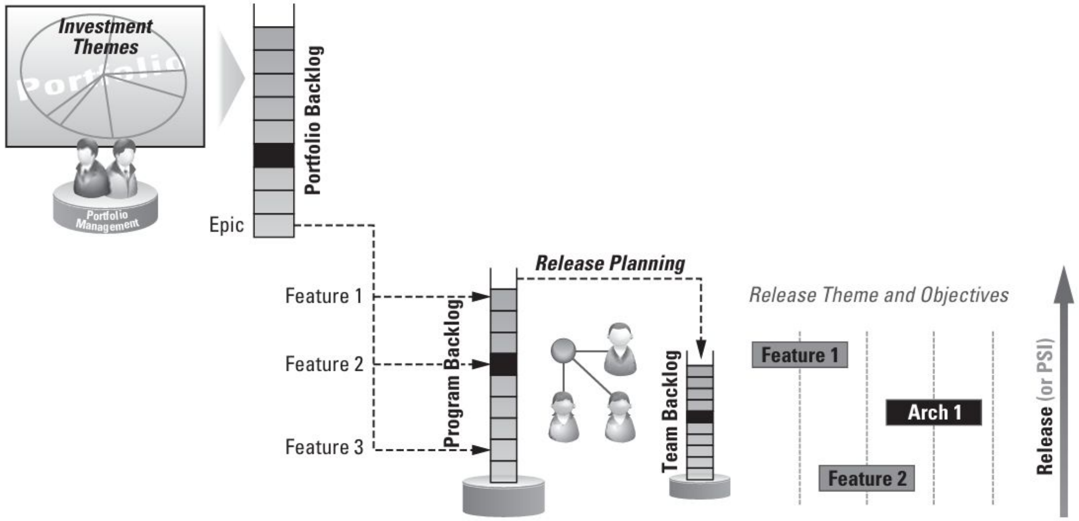
Portfolio Backlog

- ➔ Epics are identified, prioritized, estimated, and maintained in the portfolio backlog.
- ➔ Epics may be expressed in
  - ◆ bullet form
  - ◆ as a sentence or two
  - ◆ in video
  - ◆ in a prototype
  - ◆ in user interface mockups
  - ◆ or indeed in any form of expression suitable to express.



Portfolio Backlog

Prior to release planning, epics are decomposed into features, which in turn drive release planning.



Contents

5. Architectural Runway And Architectural Epic

Architectural Runway

“A system that has architectural runway contains existing or planned infrastructure sufficient to allow incorporation of current and anticipated requirements without excessive refactoring.”

Scaling Software Agility [Leffingwell 2007]

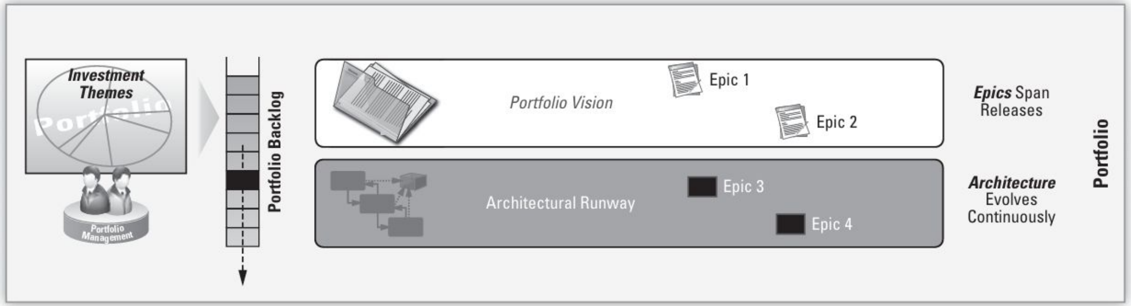
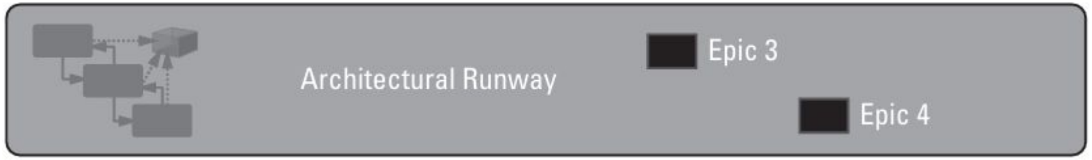


Figure 5–6 Architectural runway and architectural epics in the Big Picture

Architectural Runway

➔ In the context of the enterprise's portfolio of products and in the face of a series of shorter, incremental releases, architectural runway is the answer to a big question:

What technology initiatives need to be underway now so that we can reliably deliver a new class of features in the next year or so?



Architectural Runway

What technology initiatives need to be underway now so that we can reliably deliver a new class of features in the next year or so?

- ➔ We are not talking about side R&D projects that an enterprise may use to determine technology strategies, establish feasibility, and so on.
  - ◆ Those are localized efforts and can be managed fairly easily by the teams or the system architects.
- ➔ Rather, we are talking about large-scale changes to the code base that will be necessary to support features on the current roadmap and changes that could affect most, or even all, of the development teams.

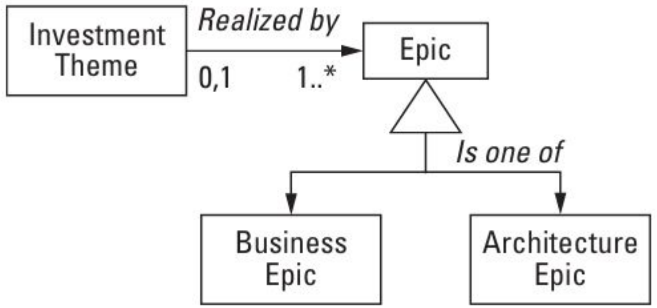
Architectural Runway

- ➔ Clearly, these are not simple refactors.
- ➔ These changes will involve significant, structural changes that could affect millions of line of code and require tens (or even hundreds) of man-years.
- ➔ And, if the enterprise wants to see it accomplished next year, or even the year after, they have to start now.
- ➔ To start now, this work has to be defined and communicated to the team, like any other major initiative, even though the end-user value may be a year or so down the road.

Architectural Epics

There Are Two Kinds Of Epics:

- ➔ **Business Epics:** These are functional or user-experience epics.
- ➔ **Architecture Epics:** These are used to implement the technological changes that must be made to significant elements of the portfolio.



Architectural Epics

Implementing Architectural Epics

Architectural epics will be implemented in the main code line, incrementally, just like any other epic.

- ➔ In so doing, development teams commit to a “do no harm” refactoring approach.
  - ◆ They implement these large-scale refactors in small increments.
- ➔ At each PSI, they commit to “do no harm” to the systems or its users, and they roll out the architectural changes piecemeal and surface the new capabilities to the users only whenever there is sufficient infrastructure to do so.
- ➔ It isn’t easy. It is agile. And it does work.

Architectural Runway: Portfolio, Program, And Team

The continuous build out and maintenance of new architectural runway is the responsibility of all mature agile teams. Failing to do so will cause one of two bad things to happen.

1. **Release dates will be missed** because large-scale, just-in-time, infrastructure refactoring adds unacceptable risk to scheduling.

2. **Failure to extend the architecture systematically** means that the teams will eventually run out of runway. New features cannot be added without major refactoring. Velocity slows. The system eventually becomes so brittle and unstable that it has to be entirely rewritten.

Architectural Runway: Portfolio, Program, And Team

Portfolio

➔ Portfolio-level architectural runway is achieved by defining, communicating, and implementing architecture epics that drive the company's technology vision.

➔ Some will require significant levels of investment and consume substantial resources.

➔ In the near term, some may even reduce the velocity of current and new feature implementations. Because failing to implement them will eventually compromise the company's position in the market, architectural epics must be visible, estimated, and planned just like any other epic.

Architectural Runway: Portfolio, Program, And Team

Program

- ➔ At the Program level, product managers, system teams, project teams, and architects translate the architectural epics into architectural features that are relevant to each release.
- ➔ They are prioritized, estimated, and resourced like any other feature.
- ➔ And, like features, each architectural initiative must also be conceptually complete at each release boundary so as to not compromise the new release.

Architectural Runway: Portfolio, Program, And Team

Team

- ➔ At the Team level, refactors and design spikes are often necessary to extend the runway, and they are prioritized along with user stories.
- ➔ In this way, architectural work is visible, accountable, and demonstrable at every iteration boundary.
- ➔ This is accomplished by agreement and collaboration with the system architects, product owners, and agile tech leads that determine what spikes need to happen and when.

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.