

Agile Software Requirements

Software Requirements Engineering – 40688

Computer Engineering department

Sharif university of technology

Fall 402

Chapter 5:

Agile Requirements For The Portfolio

Contents

- ❑ Introduction to the Portfolio level
- ❑ Investment Themes
- ❑ Portfolio Management Team
- ❑ Epics and the portfolio backlog
 - Portfolio Backlog
- ❑ Epic, Features and Stories
- ❑ Architectural Runway and Architectural Epics
 - Implementing Architectural Epics
 - Architectural Runway: Portfolio, Program, and Team
- ❑ Summary

Introduction to the Portfolio level (1)

- For software enterprises of modest scope of 100 or so practitioners.
- And those that develop and manage only one or two products.
- The team model (with its user stories, tasks, and acceptance tests)
- Plus the program model (adding features and nonfunctional requirements) may be all that the teams need to manage system requirements in an agile manner.
- 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 (2)

- There is another class of enterprises
- Enterprises employing hundreds to thousands of practitioners.
- Those that have many product
- Wherein the governance and management model for new software asset development needs additional artifacts
- And still higher levels of abstraction. In the Big Picture, this is the **Portfolio level**.

Introduction to the Portfolio level (3)

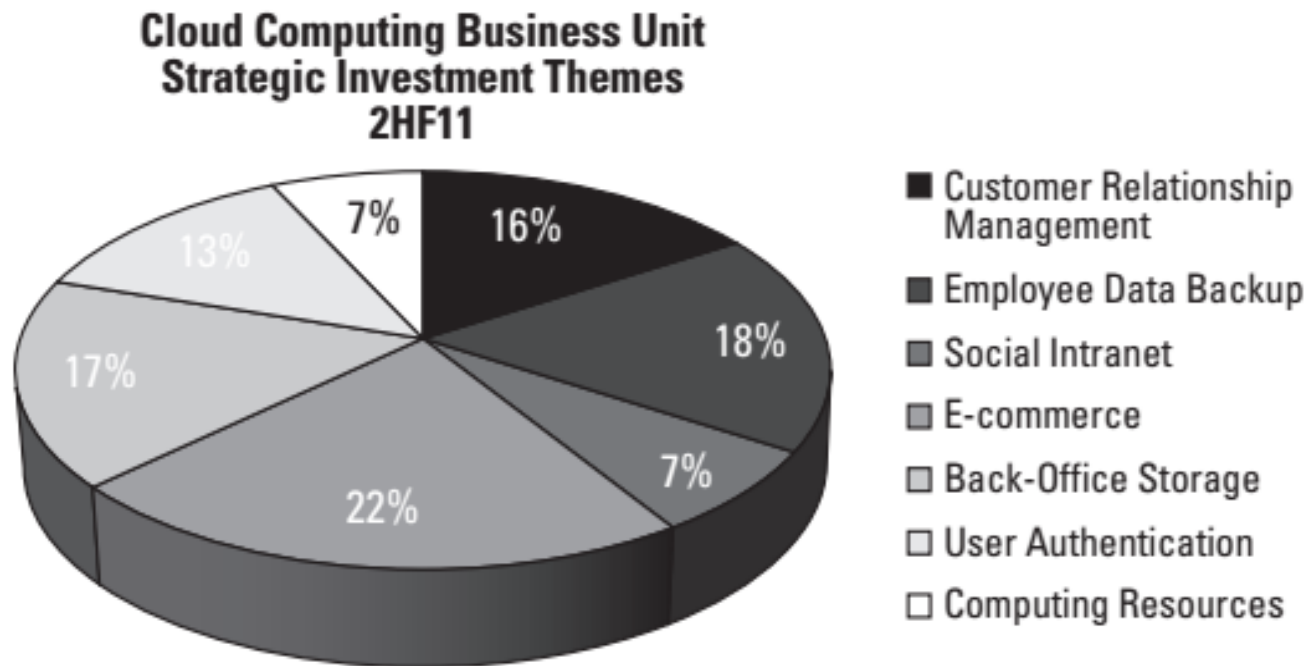
- The **Portfolio level** introduces two new artifact types:
- **Investment themes** and **epics**,
- A new backlog (the **portfolio backlog**),
- A new team (the **portfolio management team**)
- And the container concepts of **portfolio vision**.
- And **architectural runway**.

Investment Themes (1)

- Named as Product themes
- **The set of initiatives** that drive the enterprise's investment.
- In systems, products, applications, and services.
- Represent key product or service value propositions that provide differentiation and competitive advantage.
- The set of strategic investment themes for an enterprise, or business unit within an enterprise, establishes the **relative investment objectives** for the entity.

Investment Themes (2)

An example



Portfolio Management Team (1)

- The derivation of these decisions is the responsibility of the **portfolio management function**.
- 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 (2)

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

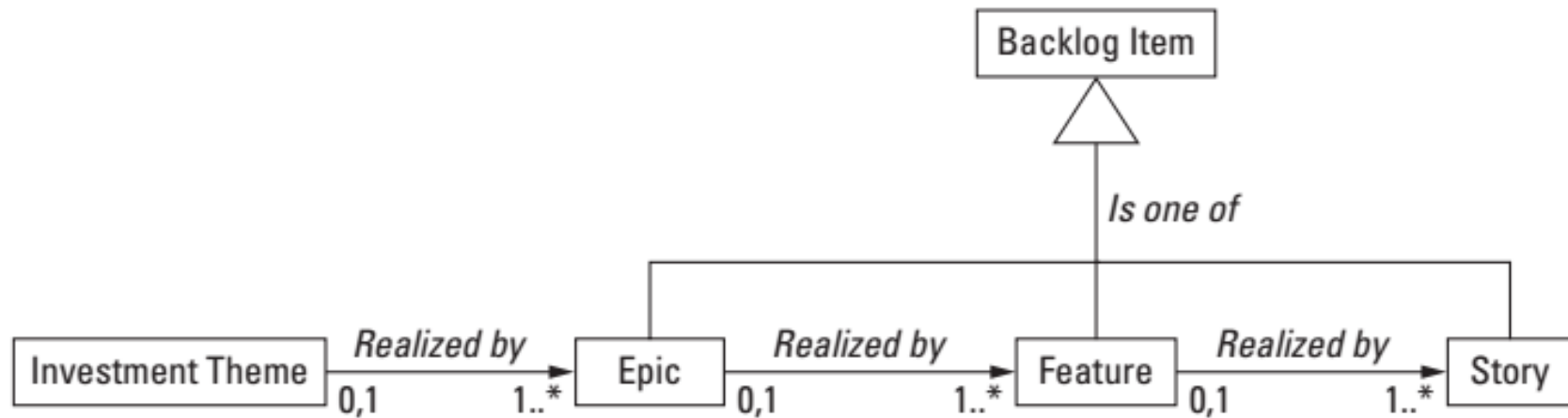
- 1) Investment in **existing** product offerings enhancements, support, and maintenance.
- 2) Investment in **new** products and services—products that will enhance revenue and/or gain new market share in the current or near-term budget period.
- 3) Investment in **futures**—advanced product and service offerings that require investment today but will not contribute to revenue until outlying years.
- 4) **Reducing** investment (sunset strategy) for existing offers that are nearing the end of their useful life.

Epics (1)

Epics are:

- **Large-scale development initiatives** that realize the value of investment themes.
- **The highest-level requirements** artifact that we will use to coordinate development.
- In the requirements model, they sit **between investment themes and features**.

Epics (2)

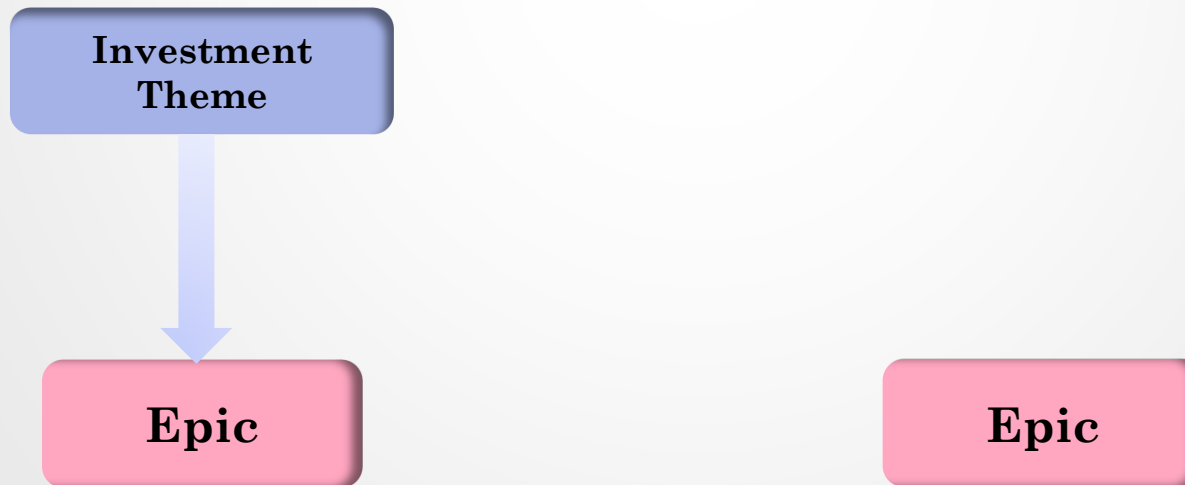


Investment themes and epics in the requirements information model

Epics (3)

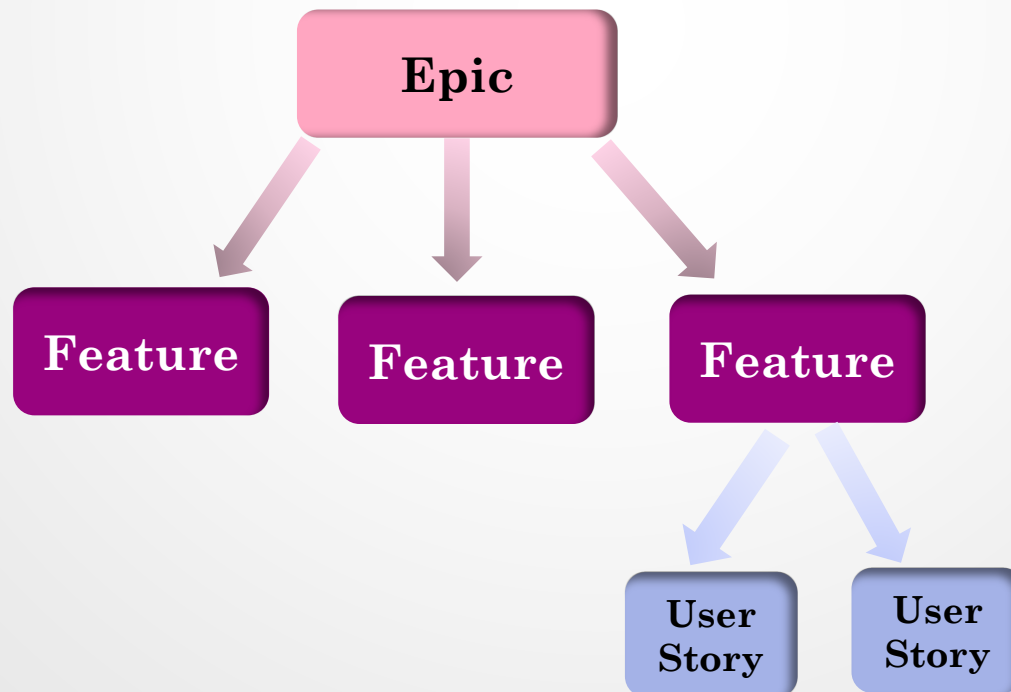
Epics are typically driven (**parented by**) investment themes.

But some epics can be **independent** (they do not require a parent in order to exist).



Epics (4)

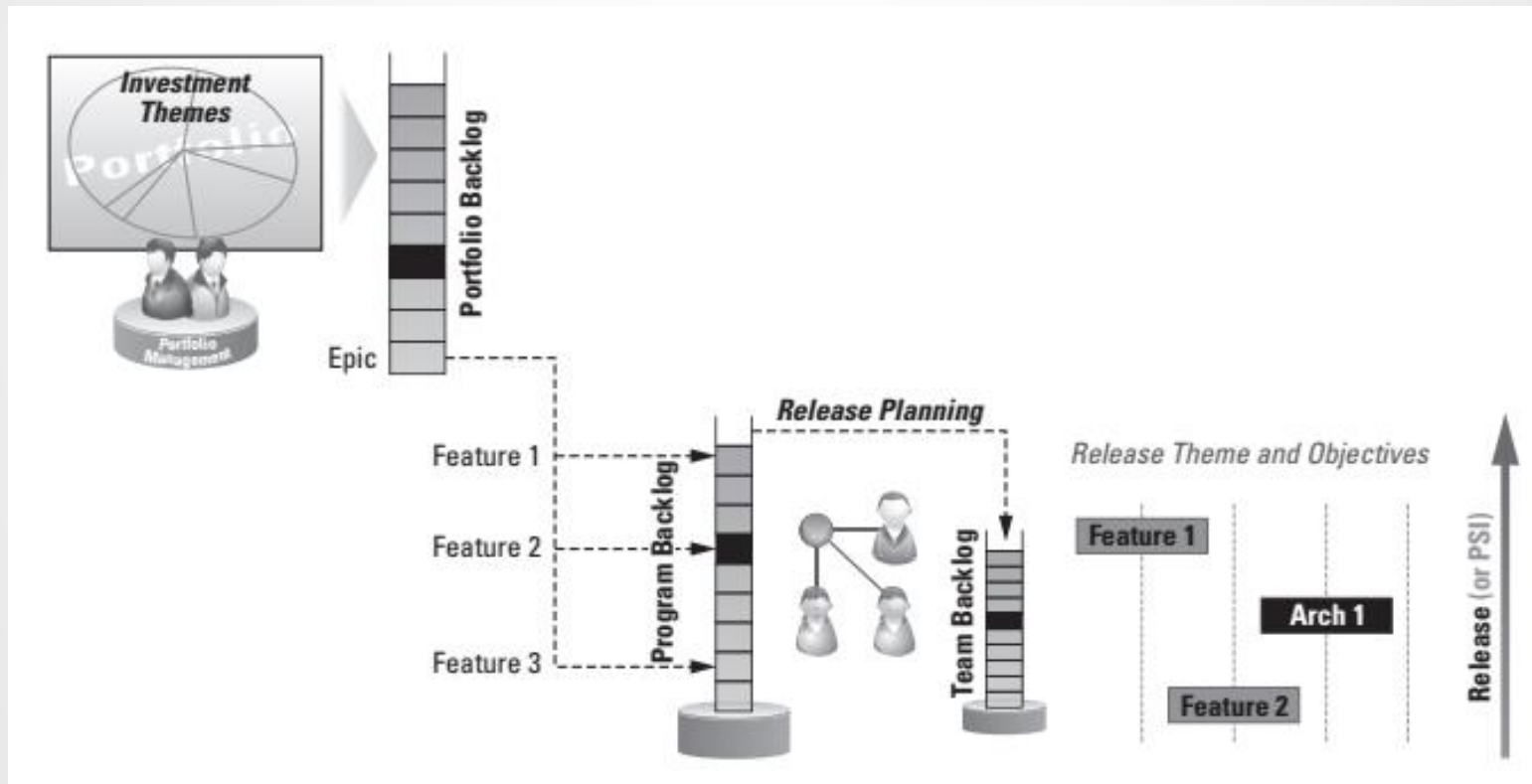
Epics are not implemented directly. Instead, they are broken into features, which, in turn, are broken into user stories, which are the primitives used by the teams for actual coding and testing.



Epics (5)

Epics are not directly testable. Instead, **they are tested by the acceptance tests** associated with the features and stories that implement them.

Portfolio Backlog

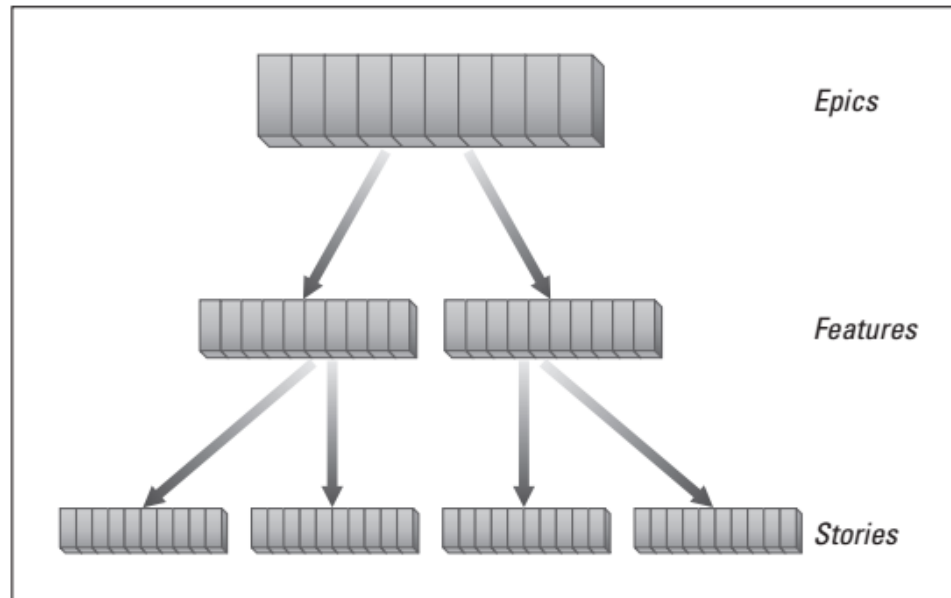


A Hierarchical view

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.



Architectural Runway And Architectural Epics(1)

- A system that has architectural runway contains **existing or planned infrastructure**.
- **This is sufficient to allow incorporation of current and anticipated requirements** without excessive refactoring.
- 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 And Architectural Epics(2)

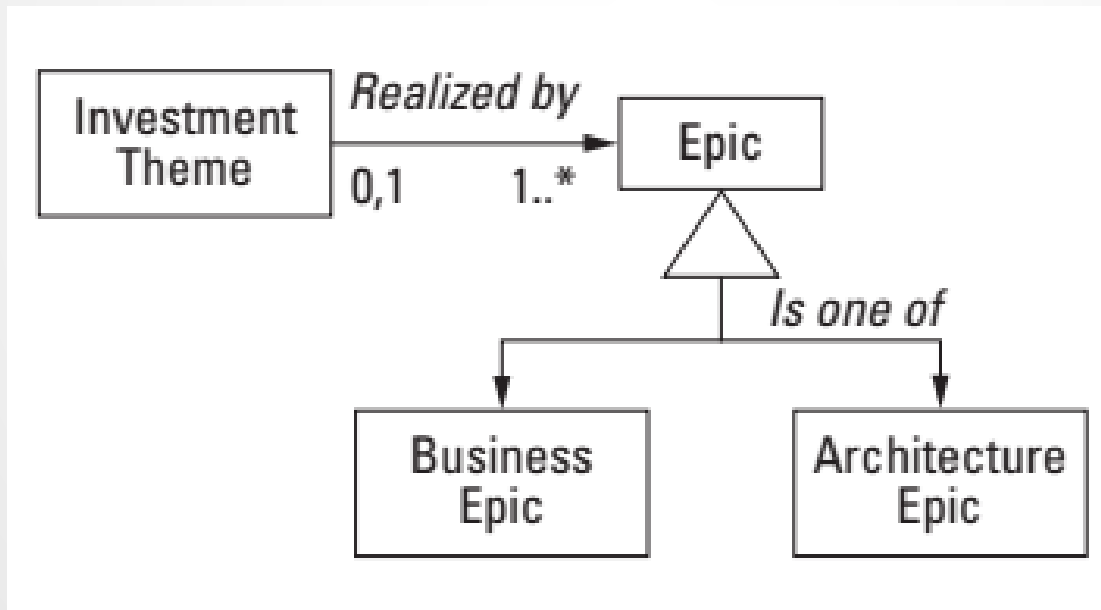
- We are talking **about large-scale changes**
- These changes are on the code base
- That will be necessary to support features on the current roadmap and changes
- That could affect most, or even all, of the development teams.

Some Examples

- Implement a common install, licensing, and user authentication model across each product in the suite.
- Convert the transaction server to a SOA-based architecture.
- Redesign the operating system to support symmetrical multiprocessing.
- 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

Business and Architecture epics

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



Architectural Runway (1): 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.

Architectural Runway (2): Portfolio, Program, and Team

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 (3): Portfolio

- Achieved by **defining, communicating, and implementing architecture epics.**
- 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 (4): 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 (5): Team

- At the Team level, refactors and design spikes are often necessary to extend the runway,
- And they are prioritized along with user stories.

Summary of Requirements

