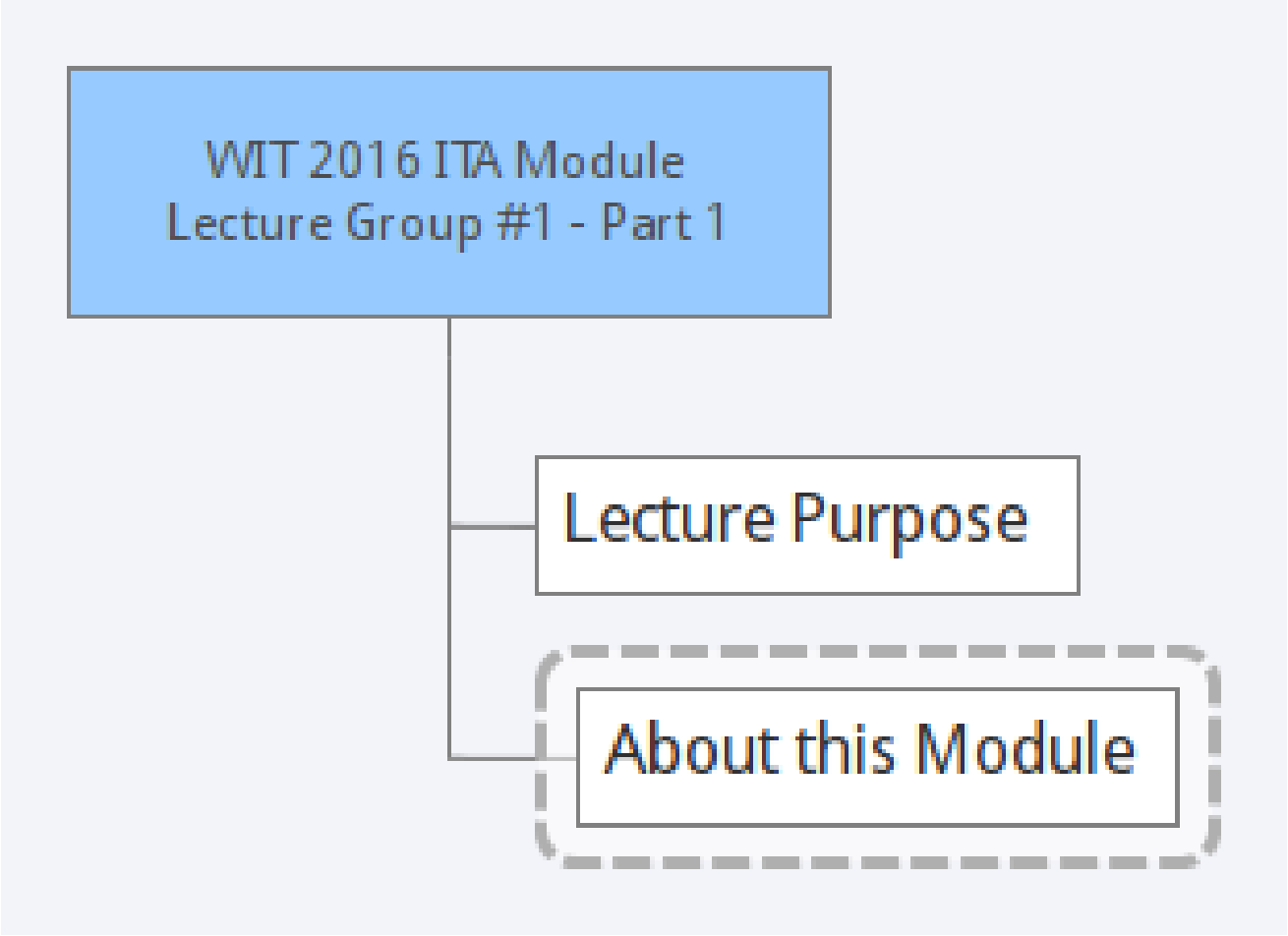
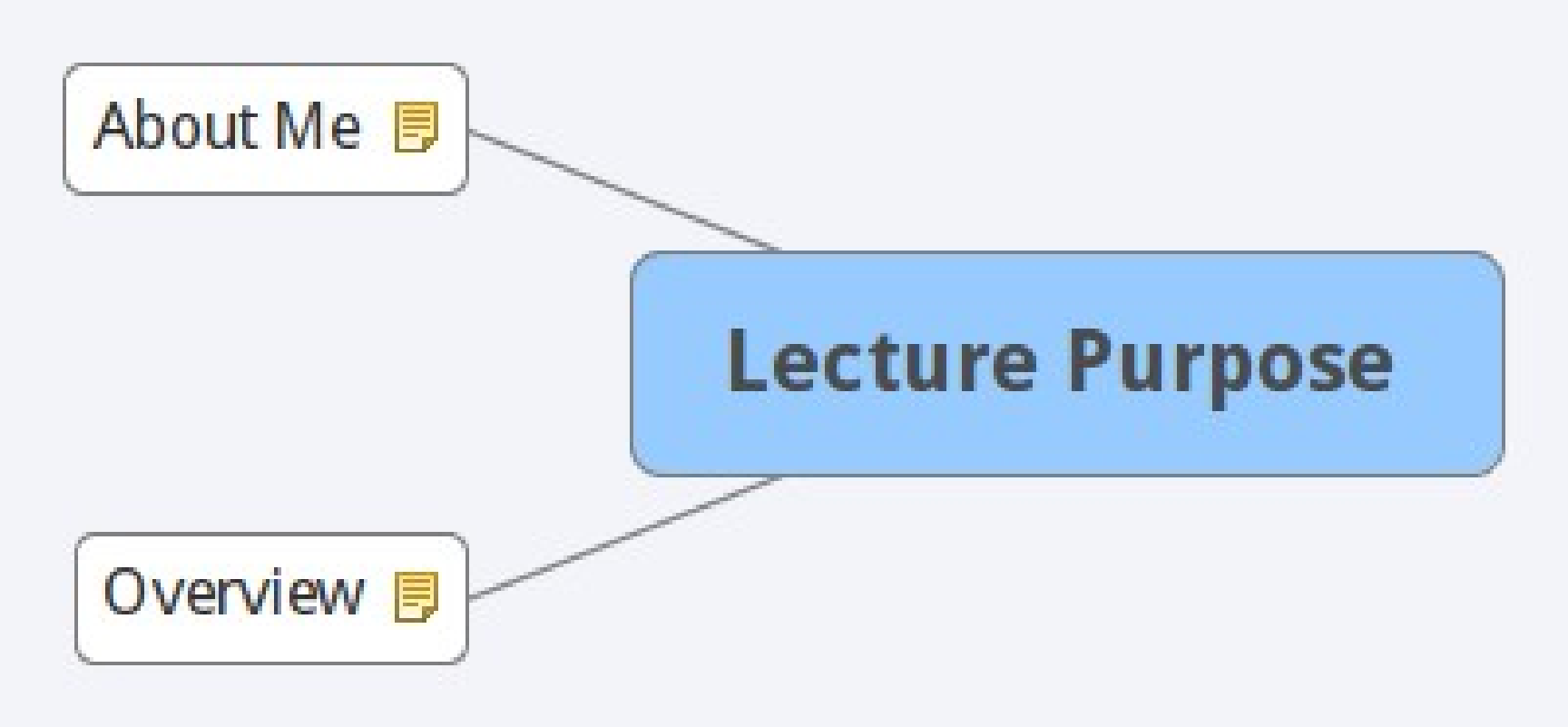
WIT 2016 ITA Module

Lecture Group #1 - Part 1

Lecture Group #1 - Part 1



Lecture Purpose



Overview

The purpose of this Section is to familiarize you with IT Architecture.

We will explain the need for ITA Styles and Patterns used to build Enterprise Software Solutions (ESS).

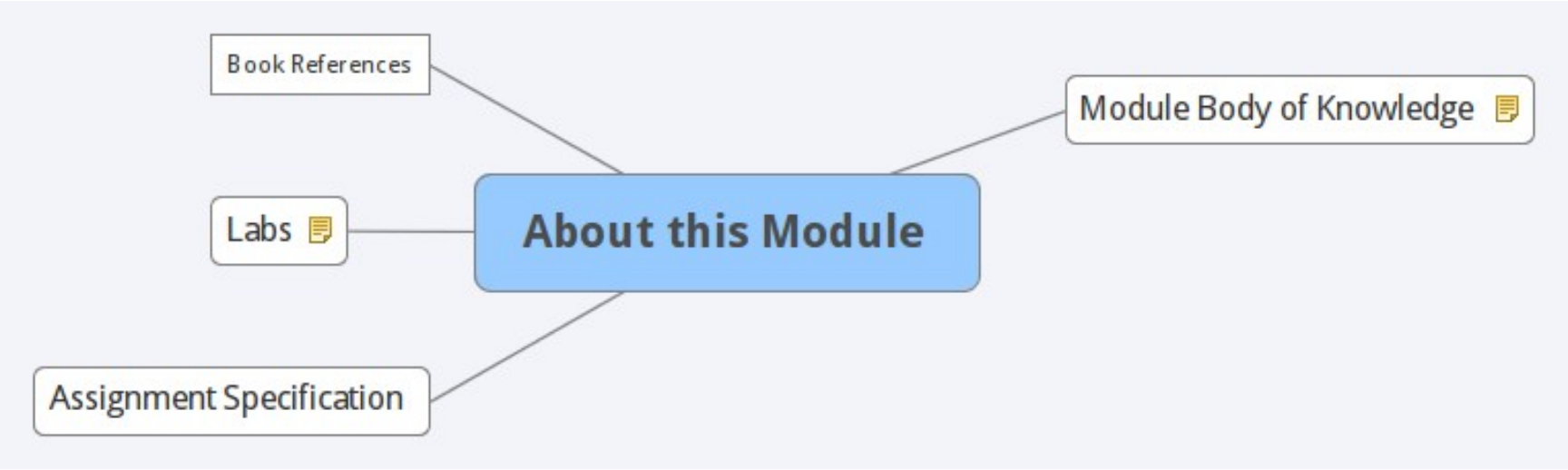
We'll review the course structure and objectives for the next few weeks.

I will brief you about what is expected from you as the Module unfolds.

About Me

The Subject of this Module.
My Subject Matter of Expertise.
Industry domain experience.
For questions or clarifications.
How to reach me (papeiclier@gmail.com). Response time to expect.
Lecture materials provided.
Access to Lectures, Labs.
Feedback.

About this Module



Module Body of Knowledge

Day 1 objective: Deep understanding of Enterprise Application Architecture

Day 2 objective: Know Tools & Techniques used by Architects for Architecture Design

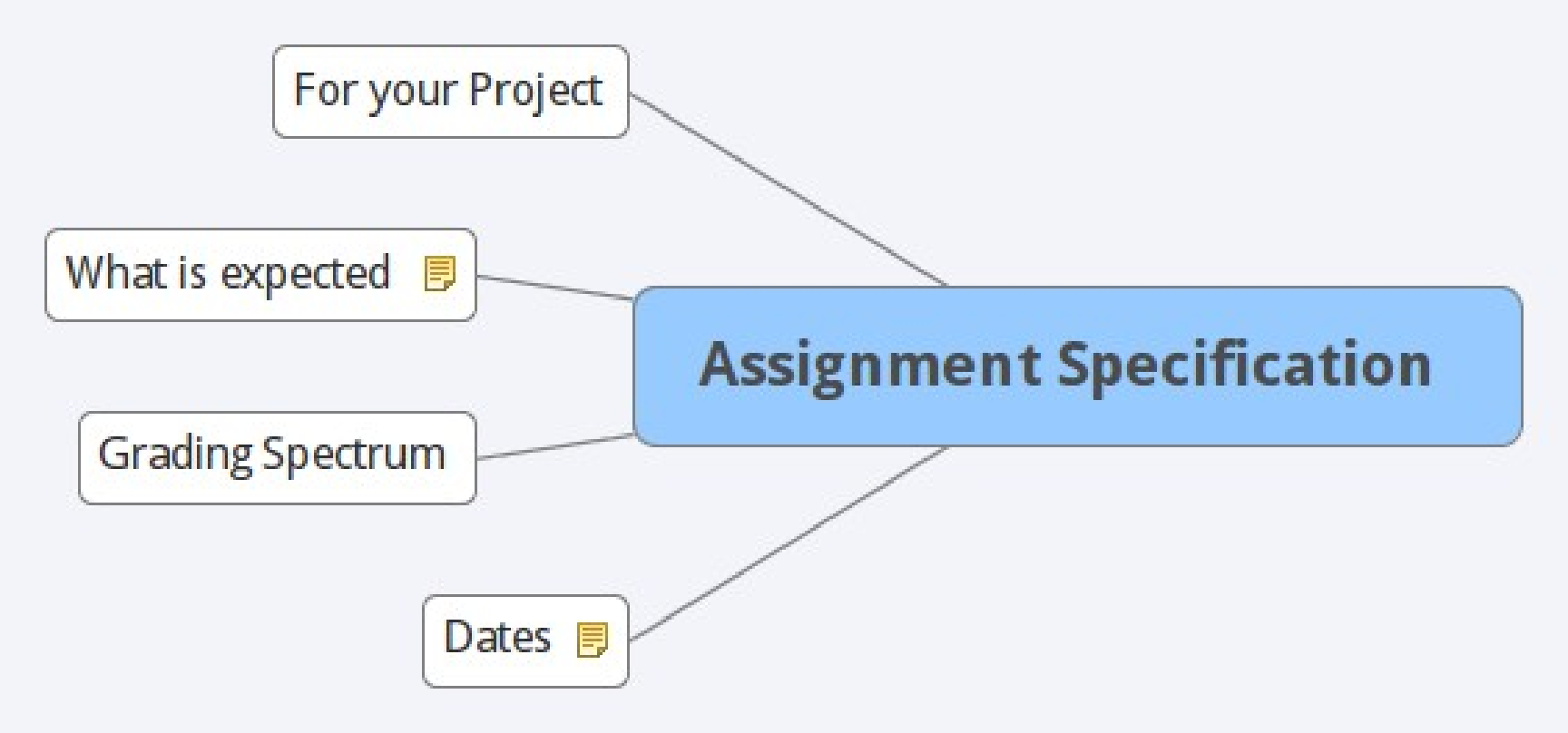
Day 3 objective: Know how to Model Architecture Solution using Archimate 2.1

Day 4 objective: Insight on a catalog of useful Architectural Styles

Day 5 objective: Insight on a catalog of useful Architecture Patterns

Day 6 objective: List of key Design Practices and Principles Architects follow

Assignment Specification



Dates

Final Deadline:

- Thursday May 12th - 11pm.

Architecture Review:

- Friday May 13th - 30 Minutes,
- schedule to be published on Moodle.

Candidate assignment rubric

Standard	Features	Patterns	Architectural Context	
Baseline	track & log activities / basic stats	3-4 Patterns. Single Catalogue	design artifacts traceable to capabilities	2-3 Viewpoints modeled. Architecture pattern visible in model. Archimate
	Colect Data to:			
Good	reports / progress insight / dashboards	4-5 Patterns. Single Catalogue	demonstrate no functional gaps in architecture	3 Viewpoints mapped. Work product feature-complete. Archimate
	Use Data to:			
Excellent	connect to friends / compare workouts	6-7 Patterns. Multiple Catalogues	solution design answers architecture requirements	3+ Visible design qualities of architecture style. Fitness for purpose proven. TOGAF AD
	Augment Data to:			
Outstanding	prescribe workouts / build training plans / live coaching	8+ Patterns. Multiple Catalogues + integrated patterns	design perspectives traceable to architecture constraints	3+ architectural trade-offs explained. TOGAF AD

What is expected

A Solution architecture work-product,

a tangible deliverable: based on the TOGAF 9.1 Architecture Description (AD) document template

expressing the architecture of an application/system from the perspective of specific system concerns

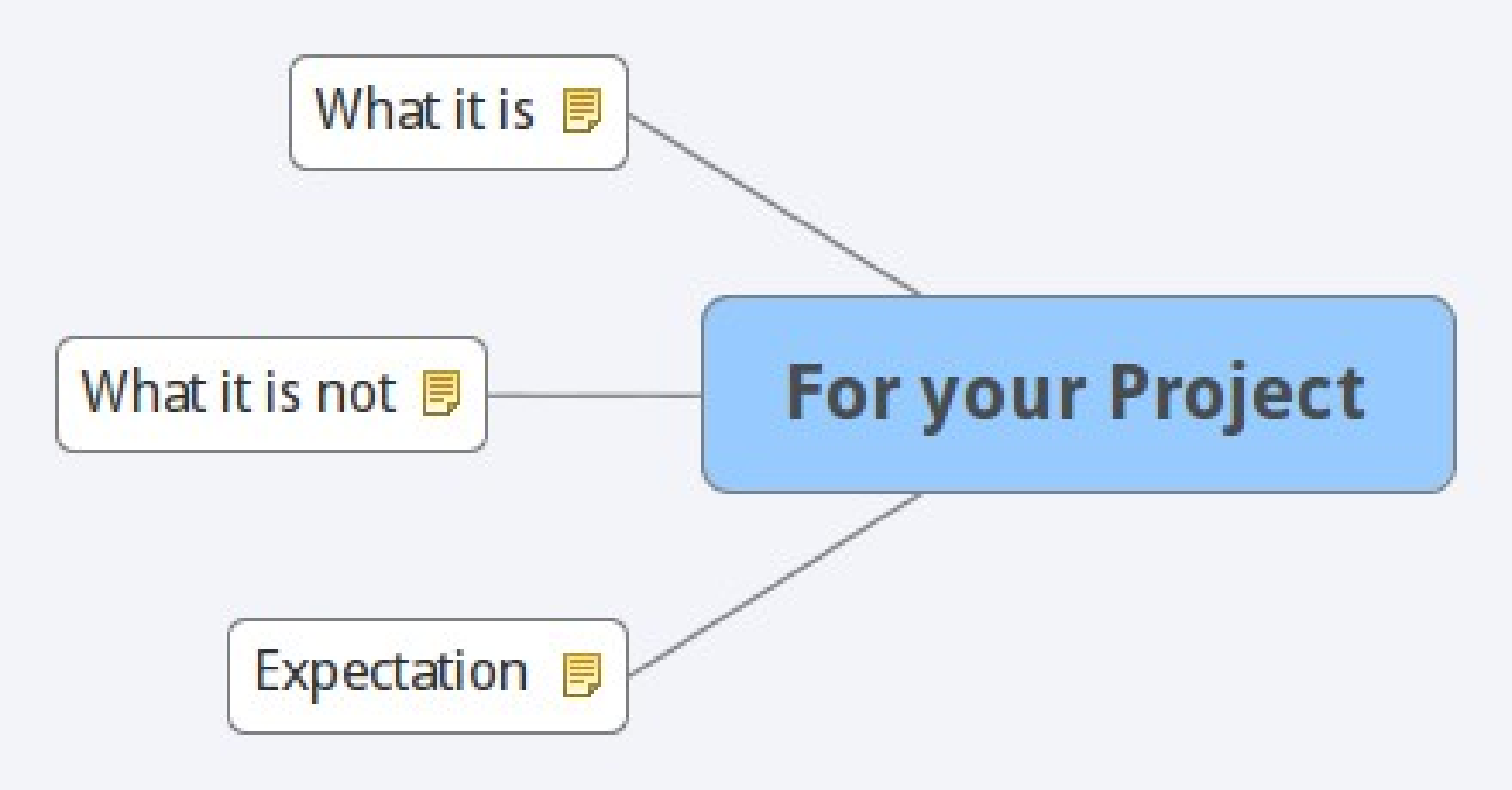
Answering to architectural requirements, linking to solution objective.

Composed of models designing Views of your solution using the Archimate 2.1 modeling notation.

Making use of at east one architecture style and patterns.

Demonstrating traceability between components, completeness of design, and quality properties of the architecture.

For your Project



Expectation

You can re-use the fictive scenario of RunKeeper.com.

You are free to re-architect the feature covered in the Labs

HOWEVER, is you decide to do so, you must use a different Architecture Style/Patterns than what is used in the Labs

...i.e. go through a DIFFERENT DESIGN PATH to the path used during our labs.

For example, you can re-architect the new RunKeeper feature we will build with a different architecture style/patterns than the ones used in the Lab.

...or you certainly can architect another new feature for RunKeeper using an Architecture Style & related Patterns of your choice (including the Style/Patterns used during the Lab)

...or shoot for different industry scenario altogether of course, using an Architecture Style & related Patterns of your choice (including the Style/Patterns used during the Lab).

What it is not

Not an extensive client-side design

Not an server-side design

Less to no coding artifacts expected in this Module

Not code-oriented work

What it is

Design/Modeling-oriented work (using the Archimate 2.1 language semantics)

Architect for a solution to a problem or goal end-to-end

Smallest artifact unit in this Module is View Modeling

Looks for proving the quality properties of your architecture

Ultimately looking at the extensibility of your solution

Labs

Lab 1 objective: Become familiar with a Architecture Design Modeling environment

From Lab 2 to 5: Model the Architecture of a new feature of RunKeeper.com

Lab 2 objective: Domain driven-design of our new RunKeeper.com feature

Lab 3 objective: Select and Apply an Architectural Style for our new RunKeeper.com feature

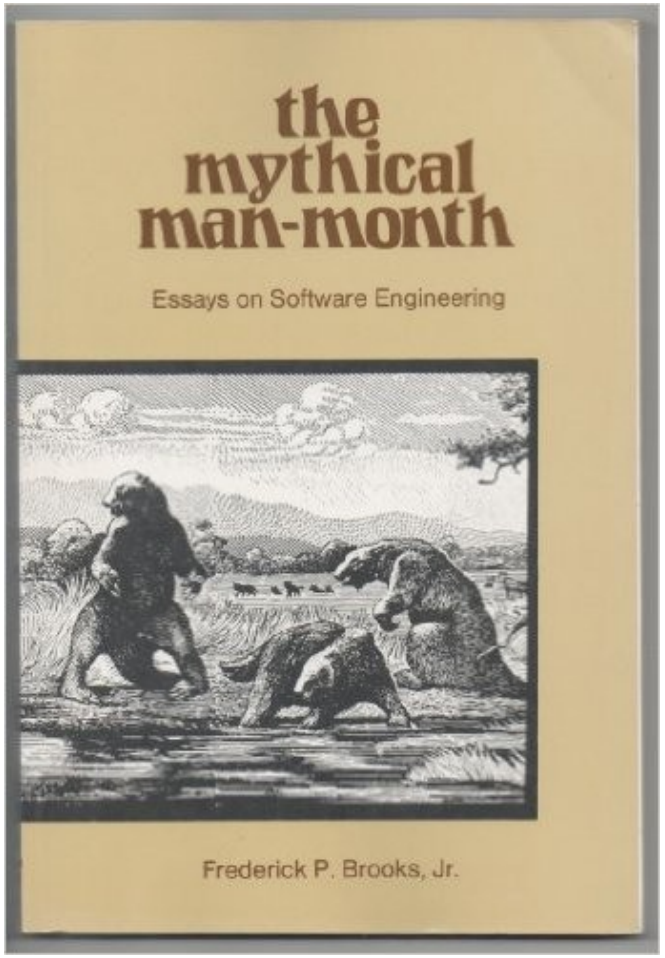
Lab 4 objective: Select and Apply an Architectural Pattern for our new RunKeeper.com feature

Lab 5 (optional): Build a TOGAF 9.1 Architecture Description deliverable (AD)

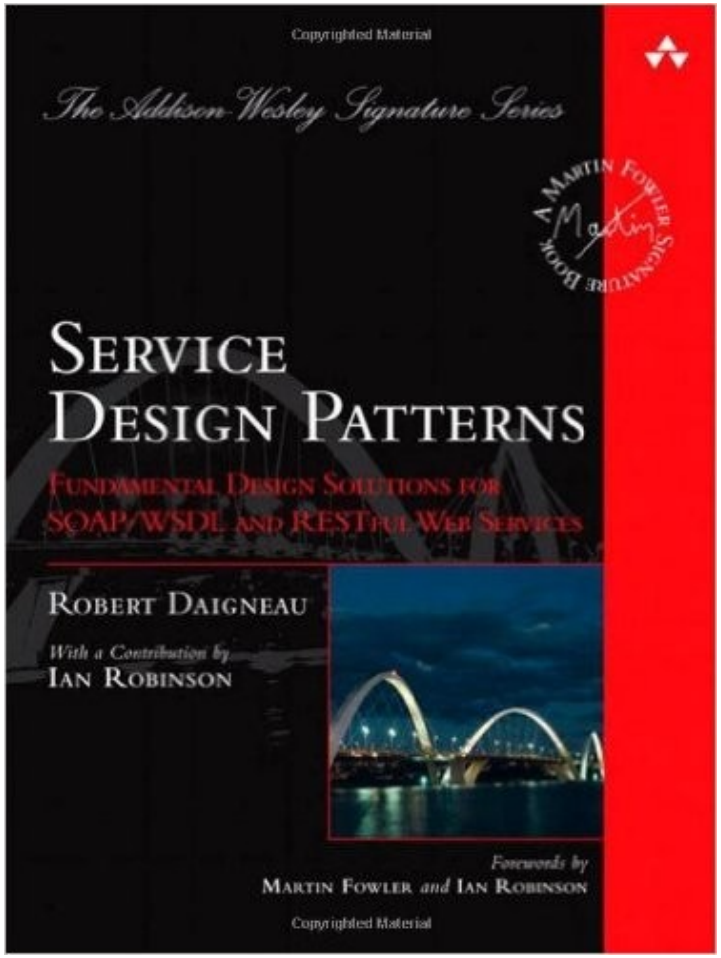
Book References

The Mythical Man-Month

- Brooks insists that there is no one silver bullet -- "there is no single development, in either technology or management technique, which by itself promises even one order of magnitude [tenfold] improvement within a decade in productivity, in reliability, in simplicity."
- The argument relies on the distinction between accidental complexity and essential complexity.
- When embarking on a second system, an engineer should be mindful that they are susceptible to over-engineering it.
- Architects can act on the user's behalf, decides what goes in the system and what stays out to ensure a user-friendly system that has conceptual integrity.

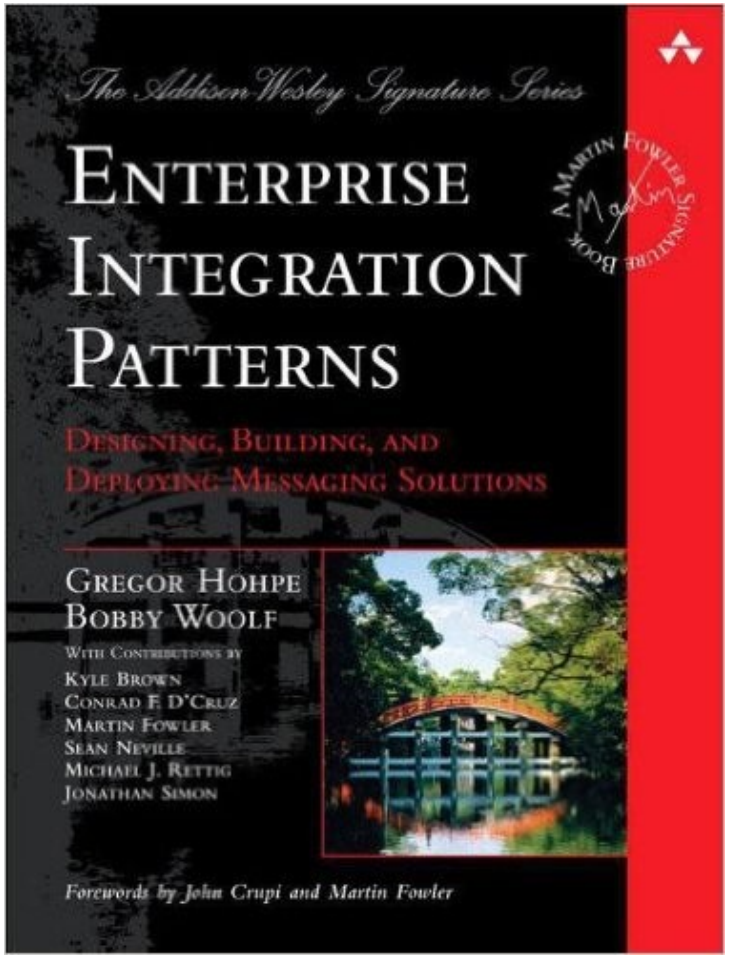


Service Design Patterns



- Expresses design solutions for web services that follow the REST architectural style or leverage the SOAP/WSDL specifications.
- Identifies the fundamental topics in web service design and lists the common design patterns.
- Explains the constituent design elements, and explore the relative strengths and trade-offs.
- Covers how solutions can adapt and change over the years.

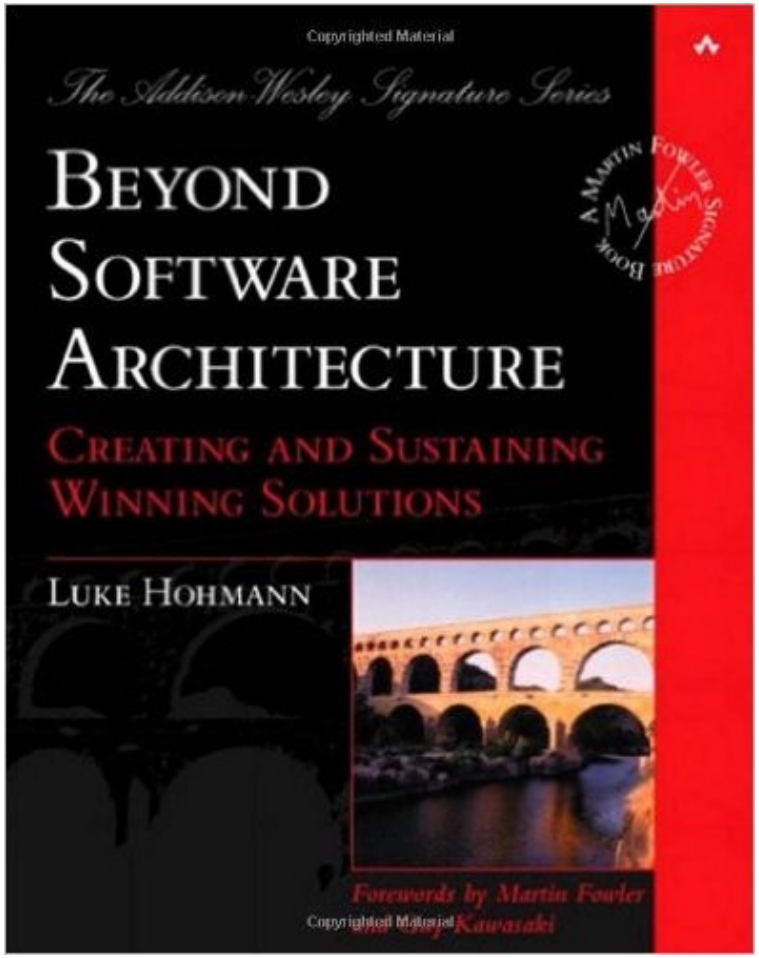
Enterprise Integration Patterns



- This book describes large-scale integration solutions across many technologies.
- It the advantages and limitations of asynchronous messaging architectures over RT architecture styles.
- Present designs to connect an application to a messaging system, determine when to send a message, how to route it to the proper destination, and how to monitor the health of a messaging system.

Beyond Software Architecture

- Provides a clear understanding of the business value of software architecture.
- Addresses how to build a software architecture that aligns with a customer's overall goals.



Peopleware

- The major issues of software development are human, not technical.
- People organization are not easy issues to solve; but if you work at understanding them in your architecture, you will maximize your chances of success.