



# Assignment Project Exam Help

SE480 Week 1 – Introduction and Overview  
<https://powcoder.com>

Steven Engelhardt

Add WeChat powcoder

Autumn 2020

# Table of Contents

1 Course Introduction

2 About Me

3 Introduction to Software Architecture

    What Is Software Architecture?

    Why Is Architecture Important?

    What Makes a Good Architect?

4 Practicing Software Architecture

    Foundations

    Structures and Views [Bass et al.]

    Aspects of Software Architecture [Vogel et al.]

    4+1 Architecture View Model [Kruchten]

    C4 Model [Brown]

# Add WeChat powcoder

5 Introduction to Quality Attributes

    Architecturally Significant Requirements

    Quality Attribute Scenarios

    Architectural Tactics

6 Wrap-Up

# Assignment Project Exam Help

- Introduction to the role of architecture and the architect in the software development cycle
- Perform design, development, and assessment activities

<https://powcoder.com>

Add WeChat powcoder

# Learning Objectives

- *Understand* the role of the Software Architect
- *Argue* the importance and role of software architecture in large-scale software systems
- *Identify* architecturally significant requirements

- *Map* a functional architecture onto a technical architecture
- *Understand* principles of good architectural design
- *Select* and *implement* appropriate architectural styles (patterns) and tactics/strategies
- *Design* preliminary software architectures which take into consideration quality goals such as reliability, performance, and scalability
- *Communicate* and *describe* software architectures both verbally and in writing
- *Evaluate* the coming attractions in software architecture research and practice

Add WeChat powcoder

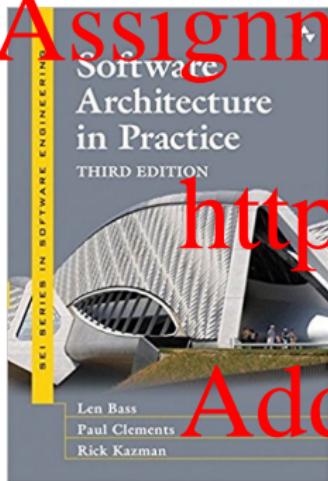
# Assignment Project Exam Help

- This course is a survey course, which means that it does not go into great depth on any given subject.
- This course is only 11 weeks long, which limits how much material we can cover

<https://powcoder.com>

Add WeChat powcoder

# Assignment Project Exam Help



Bass, Clements, and Kazman. *Software Architecture in Practice, Third Edition.* Upper Saddle River, N.J. : Addison-Wesley, 2013. ISBN: 978-0-321-81573-6.

Add WeChat powcoder

# Assignment Project Exam Help

- Email is the best way to contact me: [sengelha@depaul.edu](mailto:sengelha@depaul.edu)
- My office hours are from 16:00 - 17:30 online in Zoom. Office hours may be scheduled in [BlueStar](#)
- All notes, handouts, assignments, grades, etc. will be available through D2L. Please do not email me your assignments.
- Course homepage: <https://d2l.depaul.edu/d2l/home/767377>

<https://powcoder.com>  
Add WeChat powcoder

# Assignment Project Exam Help

- The university would like me to remind you that all class and university communications will go to your BlueM@ail address (the one ending @depaul.edu), not to your external email account (e.g. Google or Yahoo)

<https://powcoder.com>

## Add WeChat powcoder

# Assignment Project Exam Help

- SE 450
- <https://powcoder.com>
- Some Java programming will be required

Add WeChat powcoder

# Assignment Project Exam Help

- There will be 5 homework assignments and 5 quizzes
- There will be either a quiz or a homework assignment due every week
- There is some Java coding in this class, but this is not intended to be exclusively a programming class
- Assignments are generally due at shortly before class at 5:30PM, unless otherwise noted
- Late homework will *not* be accepted without strong justification

<https://powcoder.com>

Add WeChat powcoder

# Assignment Project Exam Help

- There will not be a midterm exam
- There will be a D2L-based final exam during week 11 (Thursday, November 19) – details forthcoming

Add WeChat powcoder

Quizzes	25%
Quiz 1	5%
Quiz 2	5%
Quiz 3	5%
Quiz 4	5%
Quiz 5	5%
Homework	50%
HW1	10%
HW2	10%
HW3	10%
HW4	10%
HW5	10%
Final Exam	25%
<b>Total</b>	<b>100%</b>

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Course Outline

# Assignment Project Exam Help

Wk	Date	Topics Covered	Required Reading(s)	Assignments
1	Sep 10	Introduction and overview	SAIP Ch. 1-4	
2	Sep 17	Availability, safety	SAIP Ch. 5, 12	Attendance survey due
3	Sep 24	Security, observability	SAIP Ch. 9	HW1, Quiz 1 Due
4	Oct 1	Performance, scalability	SAIP Ch. 8	Quiz 2 Due
5	Oct 8	Interoperability, modifiability, testability, deployability	SAIP Ch. 9, 7, 10	HW2 Due
6	Oct 15	Architectural patterns	SAIP Ch. 13	Quiz 3 Due
7	Oct 22	Architecture in the lifecycle	SAIP Ch. 15-22	HW3 Due
8	Oct 29	Service-oriented architecture, microservices	Supplemental	Quiz 4 Due
9	Nov 5	Cloud-native architecture, reactive architecture, event-driven architecture	Supplemental	HW4 Due
10	Nov 12	Case studies, industry examples	Supplemental	Quiz 1 Due
11	Nov 19	Final exam		HW5 Due

Add WeChat powcoder

# Assignment Project Exam Help

- SE456 – Architecture of Real-Time Systems (typically offered Winter quarter)
- SE457 – Service-Oriented Architecture (typically offered Spring quarter)

<https://powcoder.com>

Add WeChat powcoder

# Table of Contents

1 Course Introduction

2 About Me

3 Introduction to Software Architecture

    What is Software Architecture?

    Why is Architecture Important?

    What Makes a Good Architect?

4 Practicing Software Architecture

    Foundations

    Structures and Views [Bass et al.]

    Aspects of Software Architecture [Vogel et al.]

    4+1 Architecture View Model [Kruchten]

    C4 Model [Brown]

# Add WeChat powcoder

5 Introduction to Quality Attributes

    Architecturally Significant Requirements

    Quality Attribute Scenarios

    Architectural Tactics

6 Wrap-Up

# Assignment Project Exam Help

- Over 20 years of software engineering experience in varied roles
- Current role: Principal Engineer at Relativity

- Previous roles:
  - Director of Engineering at Relativity
  - Chief Data Officer at FISICO
  - 14 years at Morningstar, ending as Distinguished Engineer
  - Research Programmer at NCSA
  - Software Development Engineer at Microsoft
- Education: M.S. Computational Finance from DePaul, B.S. Computer Engineering from UIUC
- CFA Charterholder

# Table of Contents

- ① Course Introduction
- ② About Me
- ③ Introduction to Software Architecture

    What is Software Architecture?

    Why Is Architecture Important?

    What Makes a Good Architect?

- ④ Practicing Software Architecture

    Foundations

    Structures and Views [Bass et al.]

    Aspects of Software Architecture [Vogel et al.]

    4+1 Architecture View Model [Kruchten]

    C4 Model [Brown]

# Add WeChat powcoder

- ⑤ Introduction to Quality Attributes

    Architecturally Significant Requirements

    Quality Attribute Scenarios

    Architectural Tactics

- ⑥ Wrap-Up

# Table of Contents

- ① Course Introduction
- ② About Me
- ③ Introduction to Software Architecture

    What is Software Architecture?

    Why is Architecture Important?

    What Makes a Good Architect?

- ④ Practicing Software Architecture

    Foundations

    Structures and Views [Bass et al.]

    Aspects of Software Architecture [Vogel et al.]

    4+1 Architecture View Model [Kruchten]

    C4 Model [Brown]

# Add WeChat powcoder

- ⑤ Introduction to Quality Attributes

    Architecturally Significant Requirements

    Quality Attribute Scenarios

    Architectural Tactics

- ⑥ Wrap-Up

## Assignment Project Exam Help

- The software architecture of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both [BCK12]
- Software architecture is a bridge between the (often abstract) business goals and the final (concrete) resulting system [BCK12]
- Software architecture is foremost an abstraction of a system that selects certain details and suppresses others [BCK12]

Add WeChat powcoder

- [W]e will use ‘architecture’ as a noun to denote a set of artifacts, including documentation such as blueprints and building specifications that describe the object to be built, where the object is viewed as a set of structures... Put another way, the architecture is both a plan for the system so that the result can have the desired properties and a description of the built system. [SG09]
- Architecture is the composition of a set of architectural design decisions, including, amongst other things, the knowledge captured by those decisions and their rationale. [Bos04]
- The highest level breakdown of a system into its parts, the decisions that are hard to change; there are multiple architectures in a system; what is architecturally significant can change over a system’s lifetime; and, in the end, architecture boils down to whatever the important stuff is. [Fow02]

## Assignment Project Exam Help

- What requirements are the structuring and decisions based on?
- Which are the major logical and physical system building blocks?
- How are the system building blocks related to one another?
- What responsibilities do the system building blocks have?
- What interfaces exist to the system building blocks?
- How are the system building blocks grouped or layered?
- What are the specifications and criteria used to divide the system into building blocks?

<https://powcoder.com>

Add WeChat powcoder

# Is This a Good Architecture Diagram?

HDFS Architecture

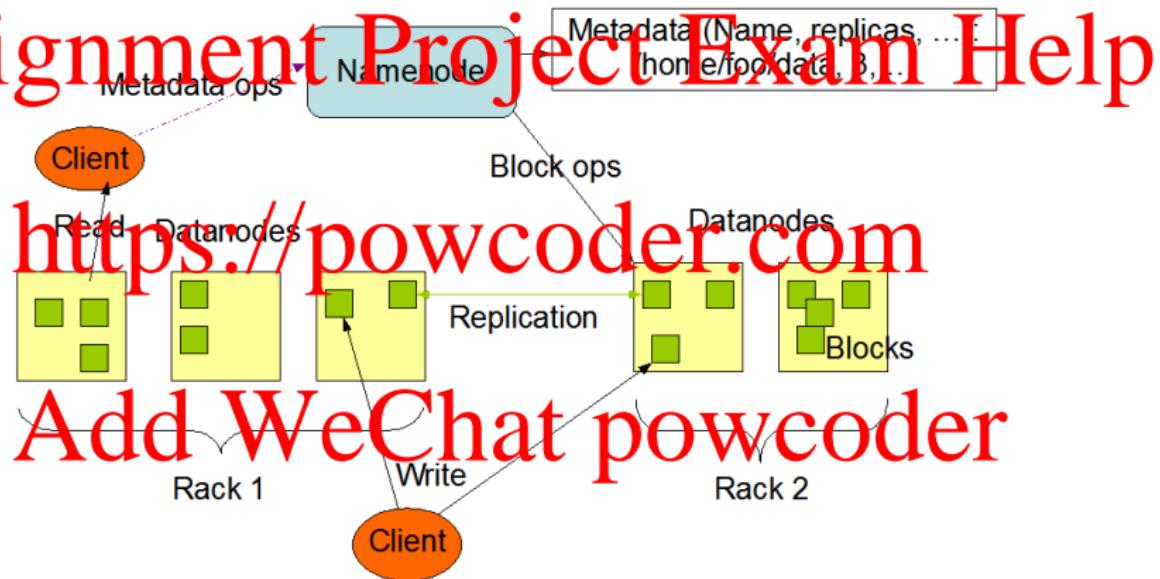


Figure: HDFS Architecture [Fou18]

## How About This?

## Figure: Spring IO Platform Architecture

# Table of Contents

- ① Course Introduction
- ② About Me
- ③ Introduction to Software Architecture

# Assignment Project Exam Help

Why Is Architecture Important?

What Makes a Good Architect?

- ④ Practicing Software Architecture

Foundations

Structures and Views [Bass et al.]

Aspects of Software Architecture [Vogel et al.]

4+1 Architecture View Model [Kruchten]

C4 Model [Brown]

# Add WeChat powcoder

- ⑤ Introduction to Quality Attributes

Architecturally Significant Requirements

Quality Attribute Scenarios

Architectural Tactics

- ⑥ Wrap-Up

## Why Architecture Matters

- An architecture will inhibit or enable a system's driving quality attributes

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

## Why Architecture Matters

- An architecture will inhibit or enable a system's driving quality attributes
  - Performance, modifiability, security, scalability, reusability, etc.
  - Allows you to reason about and manage change as the system evolves
    - Local, nonlocal, architectural changes

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Why Architecture Matters

- An architecture will inhibit or enable a system's driving quality attributes
  - Performance, modifiability, security, scalability, reusability, etc.
- Allows you to reason about and manage change as the system evolves
  - Local, nonlocal, architectural changes
- Defines a set of constraints on subsequent implementation
  - Performance budgets, restricting design alternatives to focus on implementation

Add WeChat powcoder

# Why Architecture Matters

- An architecture will inhibit or enable a system's driving quality attributes
  - Performance, modifiability, security, scalability, reusability, etc.
- Allows you to reason about and manage change as the system evolves
  - Local, nonlocal, architectural changes
- Defines a set of constraints on subsequent implementation
  - Performance budgets, restricting design alternatives to focus on implementation
- Dictates the structure of an organization, or vice versa (Conway's Law) and helps allocate work

Assignment Project Exam Help  
<https://powcoder.com>  
Add WeChat powcoder

# Why Architecture Matters

- An architecture will inhibit or enable a system's driving quality attributes
  - Performance, modifiability, security, scalability, reusability, etc.
- Allows you to reason about and manage change as the system evolves
  - Local, nonlocal, architectural changes
- Defines a set of constraints on subsequent implementation
  - Performance budgets, restricting design alternatives to focus on implementation
- Dictates the structure of an organization, or vice versa (Conway's Law) and helps allocate work
- Enables evolutionary prototyping

Assignment Project Exam Help  
<https://powcoder.com>  
Add WeChat powcoder

# Why Architecture Matters

- An architecture will inhibit or enable a system's driving quality attributes
  - Performance, modifiability, security, scalability, reusability, etc.
- Allows you to reason about and manage change as the system evolves
  - Local, nonlocal, architectural changes
- Defines a set of constraints on subsequent implementation
  - Performance budgets, restricting design alternatives to focus on implementation
- Dictates the structure of an organization, or vice versa (Conway's Law) and helps allocate work
- Enables evolutionary prototyping
- Allows architects and project managers to reason about cost and schedule

Assignment Project Exam Help  
<https://powcoder.com>

Add WeChat powcoder

# Why Architecture Matters

- An architecture will inhibit or enable a system's driving quality attributes
  - Performance, modifiability, security, scalability, reusability, etc.
  - Allows you to reason about and manage change as the system evolves
    - Local, nonlocal, architectural changes
- Defines a set of constraints on subsequent implementation
  - Performance budgets, restricting design alternatives to focus on implementation
- Dictates the structure of an organization, or vice versa (Conway's Law) and helps allocate work
- Enables evolutionary prototyping
- Allows architects and project managers to reason about cost and schedule
- Allows incorporation of independently developed components

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

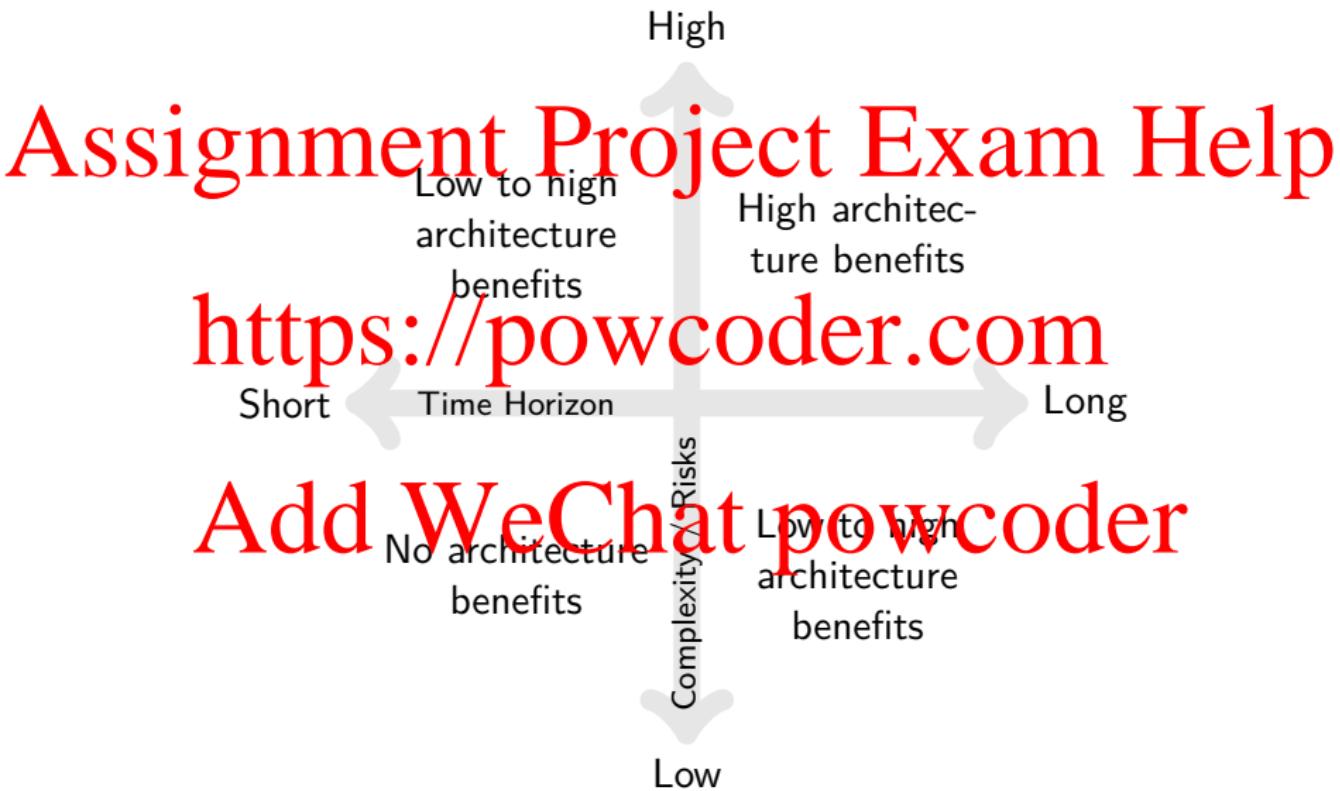
# Why Architecture Matters

- An architecture will inhibit or enable a system's driving quality attributes
  - Performance, modifiability, security, scalability, reusability, etc.
- Allows you to reason about and manage change as the system evolves
  - Local, nonlocal, architectural changes
- Defines a set of constraints on subsequent implementation
  - Performance budgets, restricting design alternatives to focus on implementation
- Dictates the structure of an organization, or vice versa (Conway's Law) and helps allocate work
- Enables evolutionary prototyping
- Allows architects and project managers to reason about cost and schedule
- Allows incorporation of independently developed components
- Foundation for training of new team members

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



# Assignment Project Exam Help

- An architecture is not inherently good or bad
- Architectures are more or less fit for some purpose
- However...

<https://powcoder.com>

Add WeChat powcoder

# Symptoms of Good Architecture

The object or system under construction has the following characteristics:[SG09]

- It has the functionality required by the customer
- It is safely buildable on the required schedule
- It performs adequately
- It is usable and safe to use
- It is secure
- It is affordable
- It conforms to legal standards
- It exhibits conceptual integrity
- It will outlast its predecessors and its competitors

I would add:

- It is well-understood by all key stakeholders
- It is easy to map new features into how they should be implemented
- It lends itself to incremental implementation
- It can be evolved over time
- It is consistent
- The number of interactions between components is kept small
- Changes rarely span multiple components
- Areas of resource contention are minimized and carefully resolved

<https://powcoder.com>

Add WeChat powcoder

# Symptoms of Poor Architecture

- Overview is missing
- Complexity runs out of control
- Planning becomes more difficult
- Early recognition of risk factors is barely possible
- Reuse of knowledge and system building blocks becomes more difficult
- Flexibility is restricted
- Maintainability becomes more difficult
- Problems with integration
- Performance is bad
- Architecture documentation is insufficient
- Learning curve for understanding the architecture is too high
- Functionality is redundant
- Development cycles are too long
- System building blocks have unnecessary dependencies
- System building blocks that cover many different responsibilities and are therefore difficult to maintain or reuse
- System building blocks whose implementation details are known in the entire system
- Numerous system building blocks have to be adapted when there is a change anywhere in the system

Assignment Project Exam Help  
<https://powcoder.com>

Add WeChat: powcoder

## Compare These Two Architectures

Assignment Project Exam Help  
<https://powcoder.com>  
Add WeChat powcoder

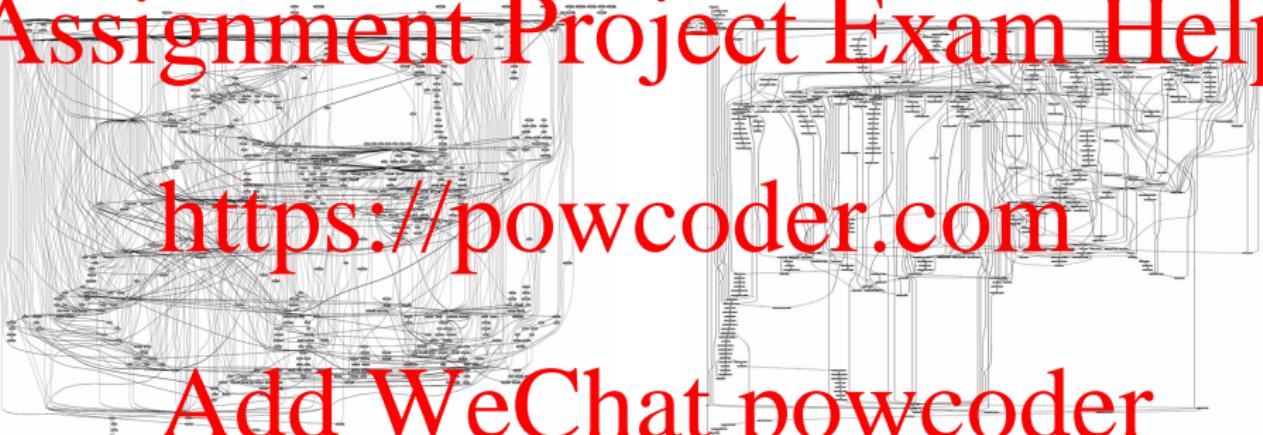


Figure: IIS System Call Trace

Figure: Apache System Call Trace



# Assignment Project Exam Help

<https://powcoder.com>

Add WeChat **powcoder**

- In 1625, the Swedish king Gustavus Adolphus ordered new warships
  - The Vasa was built in Stockholm. It was to be the mightiest warship in the world, armed with 64 guns on 2 gun decks
  - On Sunday, August 10th, 1628, the ship set sail, waddled out into Stockholm's deep water harbor, fired her guns in salute, and promptly rolled over and sank.
- <https://www.youtube.com/watch?v=0a551wa0z5E>

Figure: The Swedish warship Vasa

## Table of Contents

- ① Course Introduction
- ② About Me
- ③ Introduction to Software Architecture

# Assignment Project Exam Help

What Makes a Good Architect?

- ④ Practicing Software Architecture

Foundations  
Structures and Views [Bass et al.]

Aspects of Software Architecture [Vogel et al.]

4+1 Architecture View Model [Kruchten]

C4 Model [Brown]

# Add WeChat powcoder

- ⑤ Introduction to Quality Attributes

Architecturally Significant Requirements

Quality Attribute Scenarios

Architectural Tactics

- ⑥ Wrap-Up

- Technical credibility and programming skills

# Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

- Technical credibility and programming skills
- Problem decomposition skills

# Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

- Technical credibility and programming skills
- Problem decomposition skills
- Domain expertise

# Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

- Technical credibility and programming skills
- Problem decomposition skills
- Domain expertise
- Writing skills

# Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

## Skills Required to be a Good Architect

- Technical credibility and programming skills
- Problem decomposition skills
- Domain expertise
- Writing skills
- Business acumen

Assignment Project Exam Help  
<https://powcoder.com>

Add WeChat powcoder

- Technical credibility and programming skills
- Problem decomposition skills
- Domain expertise
- Writing skills
- Business acumen
- Mentorship and consensus-building skills

## Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

- Technical credibility and programming skills

- Problem decomposition skills

- Domain expertise

- Writing skills

- Business acumen

- Mentorship and consensus-building skills

- Leadership skills

<https://powcoder.com>

Add WeChat powcoder

- Technical credibility and programming skills

• Problem decomposition skills

- Domain expertise

- Writing skills

• Business acumen

- Mentorship and consensus-building skills

- Leadership skills

- Persuasion skills

# Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Skills Required to be a Good Architect

- Technical credibility and programming skills

- Problem decomposition skills

- Domain expertise

- Writing skills

- Business acumen

- Mentorship and consensus-building skills

- Leadership skills

- Persuasion skills

- Organizational politics awareness

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

- Technical credibility and programming skills

- Problem decomposition skills

- Domain expertise

- Writing skills

- Business acumen

- Mentorship and consensus-building skills

- Leadership skills

- Persuasion skills

- Organizational politics awareness

- Makes decisions under uncertainty

# Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

- Technical credibility and programming skills

- Problem decomposition skills

- Domain expertise

- Writing skills

- Business acumen

- Mentorship and consensus-building skills

- Leadership skills

- Persuasion skills

- Organizational politics awareness

- Makes decisions under uncertainty

- Optimism

# Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Assignment Project Exam Help

- Many senior developers, team leads, etc. act in the role of architect
- The role is frequently fulfilled by a team
- The fundamental principles and ideas of software architecture apply regardless of who fulfills them

<https://powcoder.com>

Add WeChat powcoder

# Assignment Project Exam Help

- Does the information presented so far match your expectations from before the class? In what way is it different?
- Have any of you had notable experiences with architects or architecture at your organizations, either positive or negative? What lessons did you take away from these experiences?

<https://powcoder.com>

Add WeChat powcoder

# Table of Contents

- ① Course Introduction
- ② About Me
- ③ Introduction to Software Architecture

    What is Software Architecture?

    Why is Architecture Important?

    What Makes a Good Architect?

- ④ Practicing Software Architecture

    Foundations

    Structures and Views [Bass et al.]

    Aspects of Software Architecture [Vogel et al.]

    4+1 Architectural View Model [Kruchten]

    C4 Model [Brown]

# Assignment Project Exam Help

## Add WeChat powcoder

- ⑤ Introduction to Quality Attributes

    Architecturally Significant Requirements

    Quality Attribute Scenarios

    Architectural Tactics

- ⑥ Wrap-Up

# Table of Contents

- ① Course Introduction
- ② About Me
- ③ Introduction to Software Architecture

# Assignment Project Exam Help

Why Is Architecture Important?

What Makes a Good Architect?

- ④ Practicing Software Architecture

Foundations

Structures and Views [Bass et al.]

Aspects of Software Architecture [Vogel et al.]

4+1 Architecture View Model [Kruchten]

C4 Model [Brown]

# Add WeChat powcoder

- ⑤ Introduction to Quality Attributes

Architecturally Significant Requirements

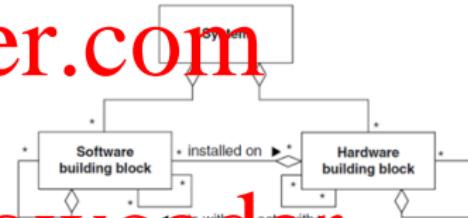
Quality Attribute Scenarios

Architectural Tactics

- ⑥ Wrap-Up

# What is a System?

- A *system* is a unit that consists of integrated software and hardware building blocks and exists for the purposes of fulfilling a functional objective.
- To achieve this objective a system communicates with its environment and must take account of the conditions defined by the environment.
- A system has a *system boundary* that separates it from its environment.
- *Emergent properties* arise from the interactions of individual system building blocks.



Add WeChat powcoder

# Example System

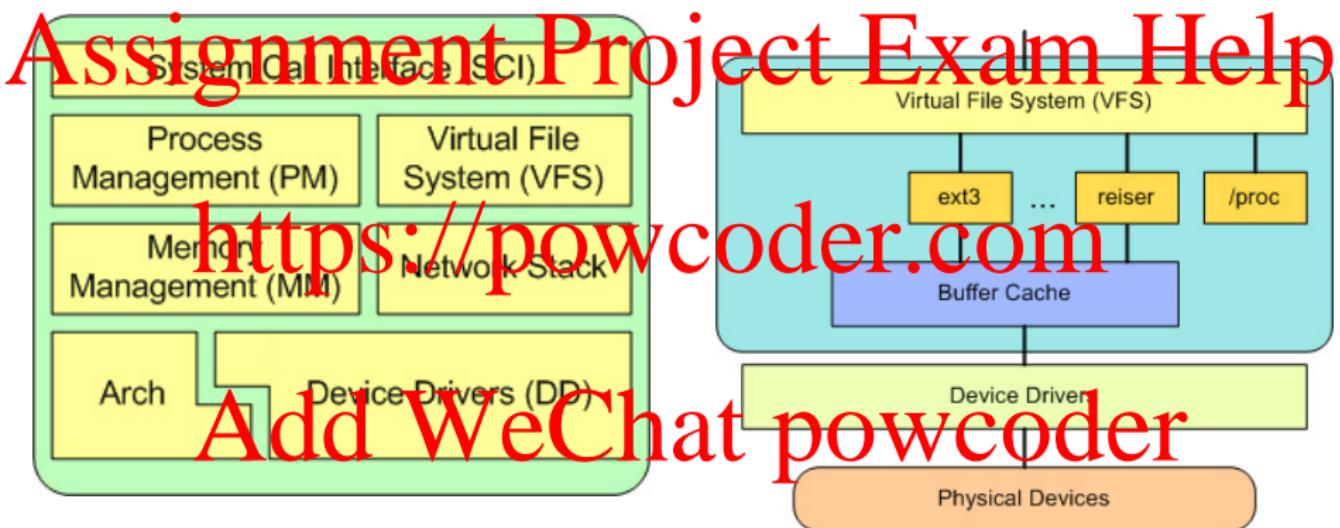


Figure: Linux Kernel Major Subsystems

Figure: Linux Kernel VFS Subsystem

# Assignment Project Exam Help

- Interaction with subsystems happens through *interfaces*, which are the mechanism to allow encapsulation
- In UML notation, open mouth requires/consumes the service lollipop provides the service

<https://powcoder.com>



Figure: UML Interface Notation

Add WeChat powcoder

- A *platform* is an underlying computer system on which application programs can run (e.g. Windows, Linux, J2E, .NET)

- A platform provides execution environments in which software building blocks are executed
- A platform is any base of technologies on which other technologies or processes are built



Add WeChat powcoder

# Assignment Project Exam Help

- The key design technique is recursive decomposition, or divide-and-conquer

- This technique can be applied in many different contexts, including:
  - Systems, subsystems, subsubsystems
  - WHO, WHAT, HOW
  - Downward allocation of KPIs
- Often decomposed subsystems can become independent systems of their own
  - SQL transaction logs
  - Blockchain

**Add WeChat powcoder**

# Assignment Project Exam Help

- *Emergent properties* are collective properties, not properties of individual parts.
- Often the nature of the interactions is more important than the identity of the parts.

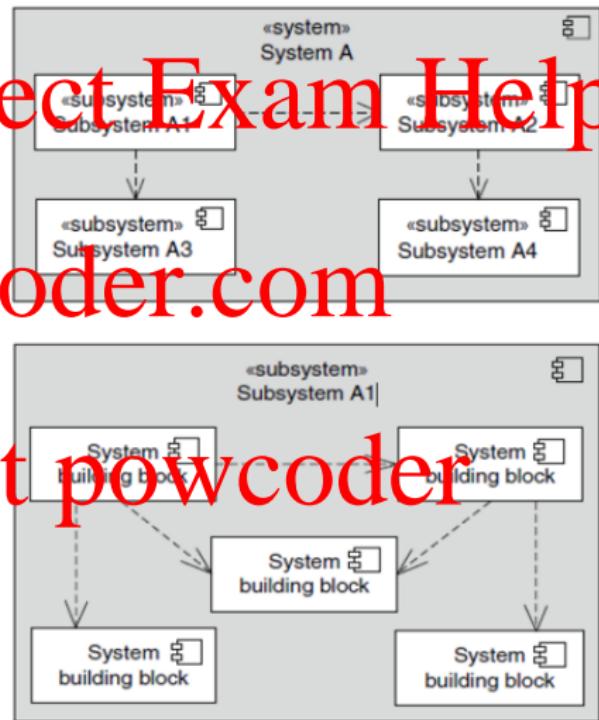
Add WeChat powcoder



Figure: A flock of birds is an emergent system, composed of individual birds.

# Holism vs. Reductionism

- The *holistic view* of systems looks at a system in its entirety. It concentrates on the emergent properties that arise through the interaction of system building blocks.
- In a *reductionism* view, the individual system building blocks are analyzed separately. This view enables a complete analysis of the behavior and function of individual system building blocks.
- The two views are complementary.



Assignment Project Exam Help  
<https://powcoder.com>

Add WeChat powcoder

# What Drives Architectural Solutions?

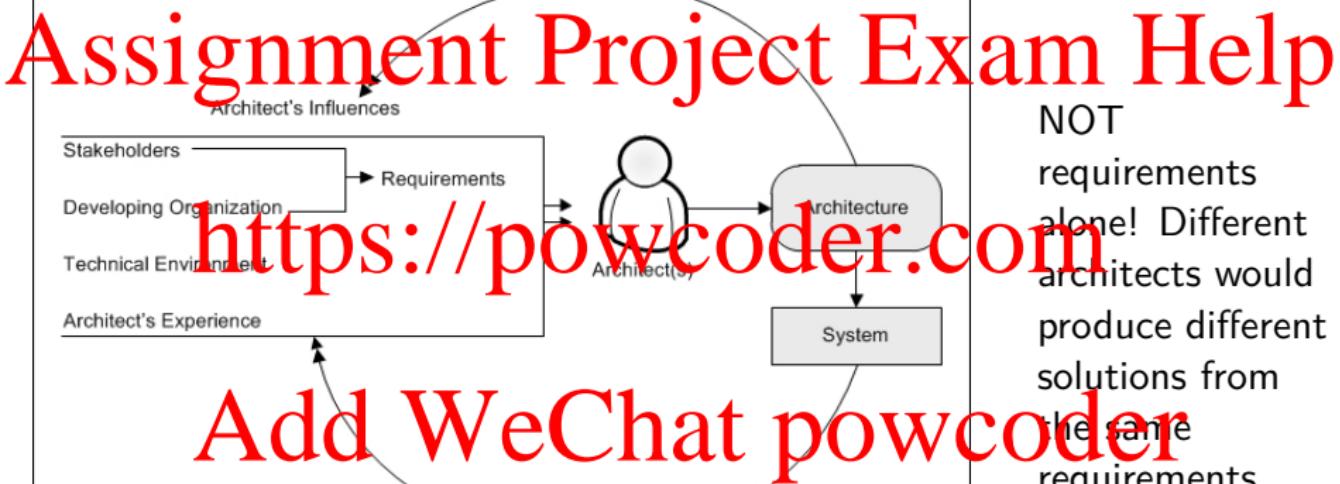


Figure: The Architecture Business Cycle

## Table of Contents

- ① Course Introduction
- ② About Me
- ③ Introduction to Software Architecture

# Assignment Project Exam Help

Why Is Architecture Important?

What Makes a Good Architect?

- ④ Practicing Software Architecture

Foundations [Fowler]

Structures and Views [Bass et al.]

Aspects of Software Architecture [Vogel et al.]

4+1 Architecture View Model [Kruchten]

C4 Model [Brown]

# Add WeChat powcoder

- ⑤ Introduction to Quality Attributes

Architecturally Significant Requirements

Quality Attribute Scenarios

Architectural Tactics

- ⑥ Wrap-Up

# Assignment Project Exam Help

- A *view* is a representation of a coherent set of architectural elements, as written by and read by system stakeholders. It consists of a representation of a set of elements and the relations among them.
- A *structure* is the set of elements itself, as they exist in software or hardware.
- Architects design structures. They document views of these structures.

<https://powcoder.com>  
Add WeChat powcoder

# Assignment Project Exam Help

<https://powcoder.com>



## Figure: Muscular subsystem (Orthopedist) A

# Add WeChat powcoder

## Views vs Structures Example

# Assignment Project Exam Help

<https://powcoder.com>



## Figure: Musculoskeletal subsystem (Orthopedist)



# WeC

Figure Skeletal  
subsystem  
(Orthopedist)

# Add WeChat powcoder

## Views vs Structures Example

# WeC

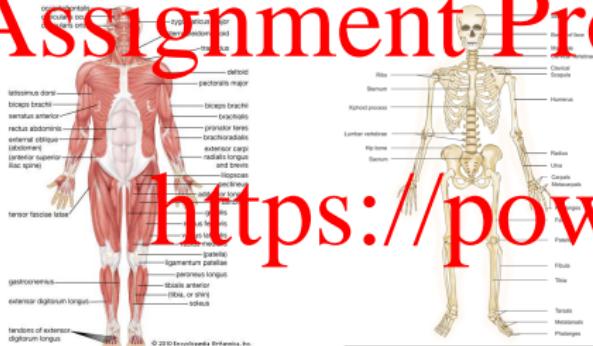
Figure Skeletal  
subsystem  
(Orthopedist)

## Views vs Structures Example

The image is a collage of four anatomical diagrams. From left to right: 1) A muscular system diagram of the human torso and arms, with labels for various muscles like latissimus dorsi, biceps brachii, rectus abdominis, etc. 2) A skeletal system diagram showing the human skeleton from the front, with labels for the skull, ribcage, spine, pelvis, and various limb bones. 3) A circulatory system diagram showing the heart and major blood vessels, with labels for the superior and inferior vena cava, aorta, and various arteries and veins. 4) An ECG tracing showing a standard 12-lead format with labels for leads I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, and V6.

# Add WeChat powcoder

## Figure: EKG (Cardiologist)



## Figure: Muscular subsystem (Orthopedist)



# Figure Skeletal subsystem (Orthopedist)



## Figure: Circulatory subsystem (Cardiologist)

# Assignment Project Exam Help

- *Module structures* embody decisions as to how the system is to be structured as a set of code or data units that have to be constructed or procured

- *Component-and-connector (C&C) structures* embody decisions as to how the system is to be structured as a set of elements that have runtime behavior (components) and interactions (connections)

- *Allocation structures* embody decisions as to how the system will relate to nonsoftware structures in its environment (such as CPUs, file systems, networks, development teams, etc.)

Add WeChat powcoder

# Assignment Project Exam Help

Module structures allow us to answer questions such as:

- What is the primary functional responsibility of each module?
- What other software elements is a module allowed to use?
- What other software does it actually use and depend on?
- What modules are related to other modules by generalization or specialization (i.e. inheritance) relationships?

<https://powcoder.com>  
Add WeChat powcoder

- *Decomposition structure*: The units are modules that are related to each other by the *is-a-submodule-of* relation, showing how modules are decomposed into smaller modules recursively until the modules are small enough to be easily understood. Used as the basis for the development project's organization.
- *Uses structure*: The units are modules that are related by the *uses* relation. A unit of software uses another if the correctness of the first requires the presence of a correctly functioning version (as opposed to a stub) of the second. Used to engineer systems that can be extended to add functionality.
- *Layer structure*: A layer is an abstract "virtual machine" that provides a cohesive set of services through a managed interface. Used to imbue a system with portability.
- *Class (or generalization) structure*: The module units in this structure are called classes. The relation is *inherits from* or *is an instance of*. Used to allow one to reason about reuse and the incremental addition of functionality.
- *Data model*: Describes the static information structure in terms of data entities and their relationships.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Assignment Project Exam Help

C&C structures allow us to answer questions such as:

- What are the major executing components and how do they interact at runtime?
- What are the major shared data stores?
- Which parts of the system are replicated?
- How does data progress through the system?
- What parts of the system can run in parallel?
- Can the system's structure change as it executes and, if so, how?

Add WeChat powcoder

# Assignment Project Exam Help

- *Service structure*: The units are services that interoperate with each other by service coordination mechanisms such as SOAP. Used to allow components to be developed anonymously and independently of each other.
- *Concurrency structure*: The units are components arranged into *logical threads* and the connectors are their communication mechanisms. Used to determine opportunities for parallelism and the locations where resource contention may occur

Add WeChat powcoder

# Assignment Project Exam Help

Allocation structures allow us to answer questions such as:

- What processor does each software element execute on?
- In what directories or files is each element stored during development, testing, and system building?
- What is the assignment of each software element to development teams?

<https://powcoder.com>

Add WeChat powcoder

## Assignment Project Exam Help

- *Deployment structure*: Shows how software is assigned to hardware processing and communication elements. Used to reason about performance, data integrity, security, and availability.

- *Implementation structure*: Shows how software elements (usually modules) are mapped to the file structure(s) in the system's development, integration, or configuration control environments. Used to manage development activities and build processes.
- *Work assignment structure*: Assigns responsibility for implementing and integrating the modules to the teams who will carry it out.

## Table of Contents

- ① Course Introduction
- ② About Me
- ③ Introduction to Software Architecture

# Assignment Project Exam Help

Why Is Architecture Important?

What Makes a Good Architect?

- ④ Practicing Software Architecture

Foundations

Structures and Views [Bass et al.]

Aspects of Software Architecture [Vogel et al.]

4+1 Architecture View Model [Kruchten]

C4 Model [Brown]

# Add WeChat powcoder

- ⑤ Introduction to Quality Attributes

Architecturally Significant Requirements

Quality Attribute Scenarios

Architectural Tactics

- ⑥ Wrap-Up

Assignment Project Exam Help

<https://powcoder.com>  
Add WeChat powcoder

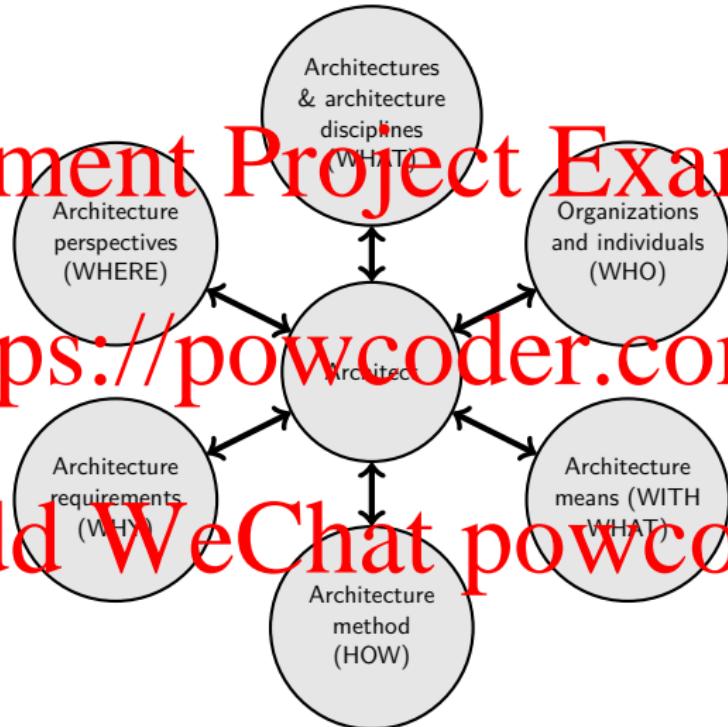


Figure: Aspects of Software Architecture [VACK11]

- Represents BASIC architectural knowledge
  - The architect must consider various architectural influencing factors such as functional and qualitative aspects, and balance them out sufficiently for the specific problem at hand.
  - Specializations in:
    - Software architecture
    - Data architecture
    - Integration architecture
    - Network architecture
    - Security architecture
    - System management architecture
    - Enterprise architecture
- Add WeChat powcoder

# Assignment Project Exam Help

:<https://powcoder.com>

:Add WeChat powcoder

- Architectural principles
- Factors, patterns, and styles
- Basic architectures
  - Dataflow architecture
  - Layered architecture
  - Middleware architecture
  - n-tier architecture
  - Rich client architecture
  - Service-oriented architecture
  - Thin client architecture
- Reference architectures
- Technologies

# Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Development organization's management

- Low cost
- Keeping people employed
- Leveraging existing corporate assets

Marketing

- Neat features
- Short time to market
- Low cost
- Parity with competing products

End user

- Behavior
- Performance
- Security
- Reliability

Maintenance organization

- Modifiability

Customer

- Low cost
- Timely delivery
- Not changed very often



Architect

# Assignment Project Exam Help

- What process do I follow?
- What do I have to do to design and implement and architecture?

• <https://powcoder.com>

- Creating the system vision
- Understanding the requirements
- Designing the architecture
- Implementing the architecture
- Communicating the architecture

• Add WeChat powcoder

# Assignment Project Exam Help

The primary motivation for architecture is not technological elegance but the *specific and long-term added value* for the business

- Ensure that the requirements placed on an IT system are *supported by the underlying architecture* of that system
- Different types of requirements must be addressed in different ways
  - Organizational requirements & constraints
  - System requirements
  - Building block requirements
  - Development time requirements
  - Runtime requirements

Add WeChat powcoder

# Assignment Project Exam Help

There are many different perspectives (views) on the same overall architecture

- People can only process  $7 \pm 2$  units of information simultaneously [Miller 1956]
- Reduce complexity by examining only one manageable part of an architecture at a time (a view)

Add WeChat powcoder



## Table of Contents

- ① Course Introduction
- ② About Me
- ③ Introduction to Software Architecture

# Assignment Project Exam Help

Why Is Architecture Important?

What Makes a Good Architect?

- ④ Practicing Software Architecture

Foundations

Structures and Views [Bass et al.]

Aspects of Software Architecture [Vogel et al.]

4+1 Architectural View Model [Kruchten]

C4 Model [Brown]

# Add WeChat powcoder

- ⑤ Introduction to Quality Attributes

Architecturally Significant Requirements

Quality Attribute Scenarios

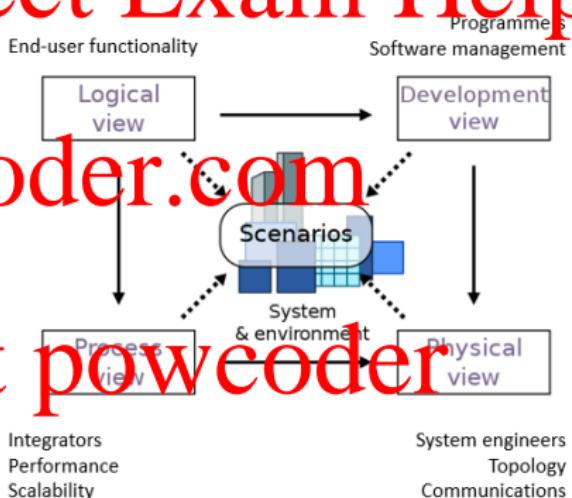
Architectural Tactics

- ⑥ Wrap-Up

# Assignment Project Exam Help

- Software architecture =  
  {Elements, Forms,  
  Rationale, Constraints}
- Formula applied separately to  
  each view
- Different styles can be applied  
  to each view

<https://powcoder.com>  
Add WeChat powcoder



## 4+1 View Model: Logical View

- Primarily supports *functional requirements* (i.e. what the system should provide to its users)
- System decomposed into a set of *key abstractions* (primarily from problem domain) in the form of objects or object classes
- Exploits abstraction, encapsulation, inheritance
- Representations: Class diagrams, E-R diagrams, state transition diagrams, state charts

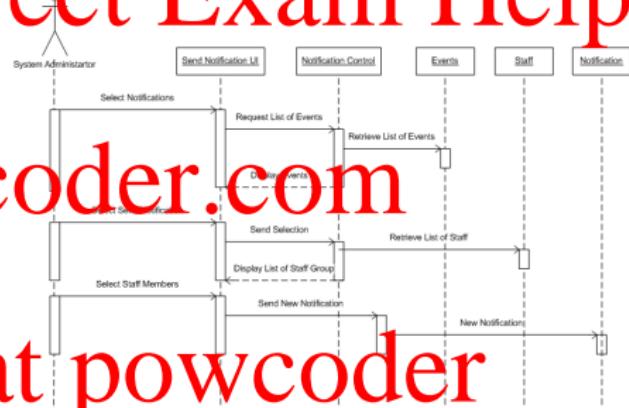


Figure: Sequence diagram

## 4+1 View Model: Process View

- Models non-functional requirements (NFRs) such as performance, availability, & fault tolerance
- A process groups tasks into an executable unit, which can be scheduled, started, recovered, reconfigured, shut down, or replicated.
- Major tasks communicate via a set of well-defined *inter-task communication mechanisms*: synchronous and asynchronous message-based communication services, remote procedure calls, etc.

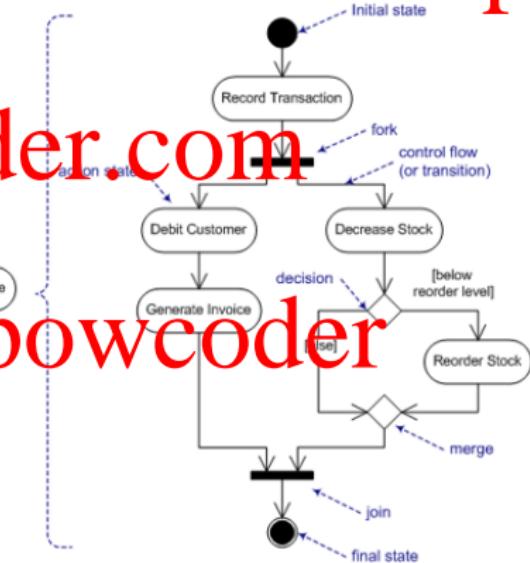


Figure: Activity diagram

# Assignment Project Exam Help

- Also referred to as *implementation view*
- Shows organization of software modules, libraries, subsystems, and units of development
- Services as an allocation view

Add WeChat powcoder

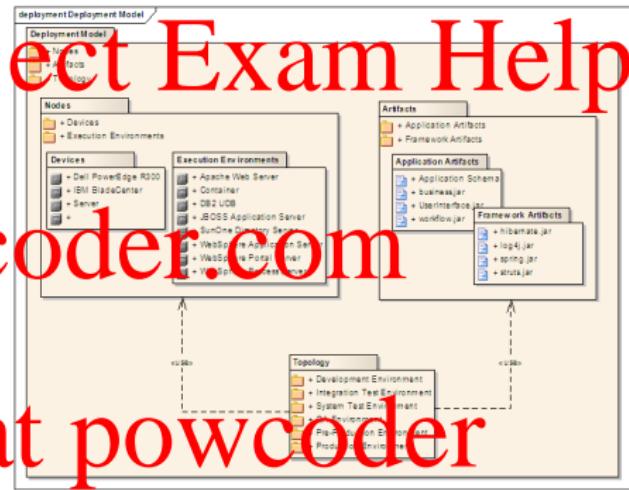
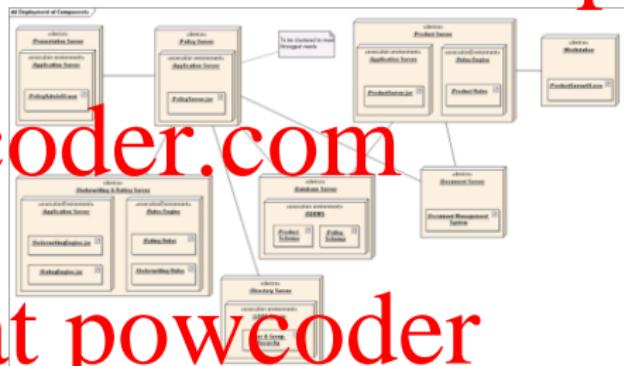


Figure: Package diagram

## 4+1 View Model: Physical View

- Also referred to as *deployment view*
  - Takes into account NFRs such as availability, reliability (fault-tolerance), performance (throughput), and scalability
  - Various elements (i.e. processes, tasks, and objects) need to be mapped onto various nodes
  - Different configurations for developing and testing, deployment, etc. Mapping needs to be flexible.



## Figure: Deployment diagram

# Table of Contents

- ① Course Introduction
- ② About Me
- ③ Introduction to Software Architecture

# Assignment Project Exam Help

Why Is Architecture Important?

What Makes a Good Architect?

- ④ Practicing Software Architecture

Foundations

Structures and Views [Bass et al.]

Aspects of Software Architecture [Vogel et al.]

4+1 Architecture View Model [Kruchten]

C4 Model [Brown]

# Add WeChat powcoder

- ⑤ Introduction to Quality Attributes

Architecturally Significant Requirements

Quality Attribute Scenarios

Architectural Tactics

- ⑥ Wrap-Up

- # Assignment Project Exam Help
- The C4 model is an “abstraction first” approach to diagramming software architecture, based upon abstractions that reflect how software architects and developers think about and build software.
  - The C4 model considers the static structures of a software system in terms of containers, components and code, all wrapped up in a system context
  - An architect produces the above four diagrams, at different levels of detail which tell different stories to different type of audiences.
  - Once you have a good understanding of the static structure, you can supplement the C4 diagrams to show other aspects.
- Add WeChat powcoder

- A **container** represents something that hosts code or data.

- In real terms, a container is something like:

- **Server-side web application:** A Java EE web application running on Apache Tomcat, an ASP.NET MVC application running on Microsoft IIS, a Ruby on Rails application running on WEBrick, a Node.js application, etc.

- **Client-side web application:** A JavaScript application running in a web browser using Angular, Backbone.JS, jQuery, etc).

- **Client-side desktop application:** A Windows desktop application written using WPF, an OS X desktop application written using Objective-C, a cross-platform desktop application written using JavaFX, etc.

- **Mobile app:** An Apple iOS app, an Android app, a Microsoft Windows Phone app, etc.

- etc.

Assignment Project Exam Help

.https://powcoder.com

Add WeChat powcoder

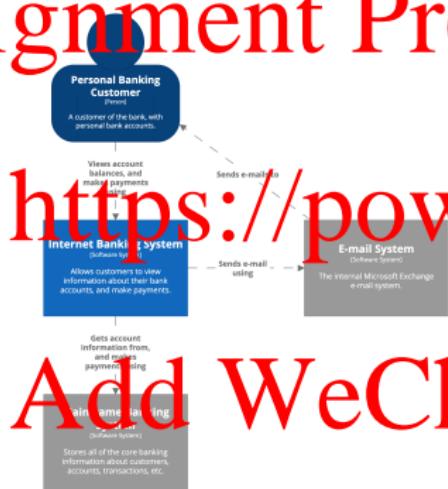
# Assignment Project Exam Help

- In this context, a **component** is simply a grouping of related functionality encapsulated behind a well-defined interface
- If you're using a language like Java or C#, the simplest way to think of a component is that it's a collection of implementation classes behind an interface
- All components inside a container typically execute in the same process space

Add WeChat powcoder

## Example C4 Architecture Diagram: Context

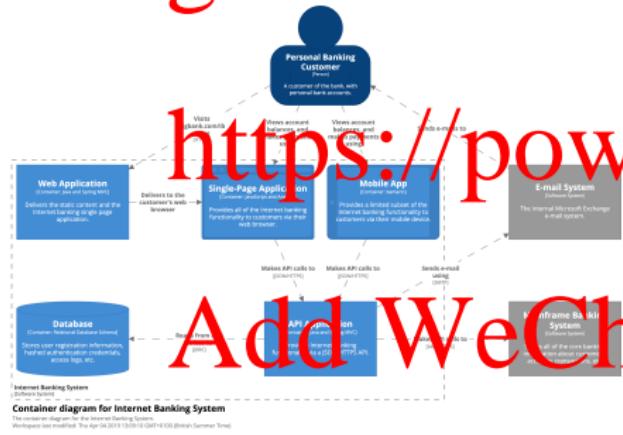
# Assignment Project Exam Help



- **Scope**: A single software system.
- **Primary elements**: The software system in scope.
- **Supporting elements**: People and software systems directly connected to the software system in scope.
- **Intended audience**: Everybody, both technical and non-technical people, inside and outside of the software development team.

## Example C4 Architecture Diagram: Container

# Assignment Project Exam Help



- Scope:** A single software system.
- Primary elements:** Containers within the software system in scope.
- Supporting elements:** People and software systems directly connected to the containers.
- Intended audience:** Technical people inside and outside of the software development team; including software architects, developers and operations/support staff.
- Notes:** This diagram says nothing about deployment scenarios, clustering, replication, failover, etc.

# Example C4 Architecture Diagram: Component

# Assignment Project Exam Help



Scope: A single container.

- Primary elements: Components within the container in scope.

- Supporting elements: Containers (within the software system in scope) plus people and software systems directly connected to the components.

Intended audience: Software architects and developers.

Add WeChat [powcoder](https://powcoder.com)

# Assignment Project Exam Help



- **Scope:** A single component.
- **Primary elements:** Code elements (e.g. classes, interfaces, objects, functions, database tables, etc) within the component in scope.
- **Intended audience:** Software architects and developers

Add WeChat powcoder

# Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



- **Scope:** An enterprise.
- **Primary elements:** People and software systems related to the enterprise in scope.
- **Intended audience:** Technical and non-technical people, inside and outside of the software development team.

## Example C4 Supplementary Diagram: Deployment



• **Scope:** A single software system.

• **Primary elements:** Deployment nodes and containers within the software system in scope.

• **Intended audience:** Technical people inside and outside of the software development team; including software architects, developers and operations/support staff.

Add WeChat powcoder

<https://powcoder.com>

# Assignment Project Exam Help

- Have you seen any of these used in practice before? Which one(s)?
- <https://powcoder.com>

Add WeChat powcoder

## Table of Contents

- ① Course Introduction
- ② About Me
- ③ Introduction to Software Architecture

# Assignment Project Exam Help

Why Is Architecture Important?

What Makes a Good Architect?

- ④ Practicing Software Architecture

Foundations

Structures and Views [Bass et al.]

Aspects of Software Architecture [Vogel et al.]

4+1 Architecture View Model [Kruchten]

C4 Model [Brown]

# Add WeChat powcoder

- ⑤ Introduction to Quality Attributes

Architecturally Significant Requirements

Quality Attribute Scenarios

Architectural Tactics

- ⑥ Wrap-Up

# Assignment Project Exam Help

- A *quality attribute* is a measurable or testable property of a system that is used to indicate how well the system satisfies the needs of its stakeholders.
- Systems are frequently redesigned not because they are functionally deficient, but because they are difficult to maintain, port, or scale; or they are too slow; or they have been compromised by hackers. [Bass et al.]

<https://powcoder.com>  
Add WeChat powcoder

# Assignment Project Exam Help

- *Functional requirements* state what the system must do, and how it must behave or react to runtime stimuli.
- *Quality attribute requirements* are qualifications of the functional requirements or of the overall product, e.g. “how fast” or “how resilient”
- *Constraints* are design decisions with zero degrees of freedom

<https://powcoder.com>  
Add WeChat powcoder

# Assignment Project Exam Help

- Functional requirements are satisfied by assigning an appropriate sequence of responsibilities through the design.
- Functional requirements do not solely determine architecture, but functionality is achieved by assigning responsibilities to architectural elements.

<https://powcoder.com>

Add WeChat powcoder

## Assignment Project Exam Help

- Quality attribute requirements are satisfied by the various structures designed into the architecture, and the behaviors and interactions of the elements that populate those structures.
  - Categories of quality attributes
    - Properties of the system at runtime (e.g. availability, performance, or usability)
    - Properties of the development of the system (e.g. modifiability, testability)
- Add WeChat powcoder

# Assignment Project Exam Help

- Constraints are satisfied by accepting them.



## Add WeChat powcoder

# Table of Contents

- ① Course Introduction
- ② About Me
- ③ Introduction to Software Architecture

# Assignment Project Exam Help

Why Is Architecture Important?

What Makes a Good Architect?

- ④ Practicing Software Architecture

Foundations

Structures and Views [Bass et al.]

Aspects of Software Architecture [Vogel et al.]

4+1 Architecture View Model [Kruchten]

C4 Model [Brown]

# Add WeChat powcoder

- ⑤ Introduction to Quality Attributes

Architecturally Significant Requirements

Quality Attribute Scenarios

Architectural Tactics

- ⑥ Wrap-Up

## Assignment Project Exam Help

- *Architecturally significant requirements* are those requirements that play an important role in determining the architecture of the system

- Technically challenging
- Technically constraining
- Central to the system's purpose

:  
<https://powcoder.com>

- The system will generally be more sensitive to changes against architecturally significant requirements
- Identifying and committing to this subset will help others understand the potential implications of change.

Add WeChat powcoder

- # Assignment Project Exam Help
- The benefit of the requirement to stakeholders: *critical, important, or useful*
  - The architectural impact of the requirement: *none, extends, or modifies*
    - Critical requirements with little or no impact on the architecture
    - Low-benefit requirements that have a big impact – perhaps eliminate?
  - The risks to be mitigated: performance, availability of a product, and suitability of a component
- <https://powcoder.com>
- Add WeChat powcoder

# Assignment Project Exam Help

The following are good examples of architecturally significant requirements:

- The system must record every modification to customer records for audit purposes.
- The system must respond within 5 seconds.
- The system must deploy on Microsoft Windows 10 and Linux.
- The system must encrypt all network traffic.
- The ATM system must dispense cash on demand to validated account holders with sufficient cleared funds.

<https://powcoder.com>

Add WeChat powcoder

# Assignment Project Exam Help

Within complex systems, quality attributes can never be achieved in isolation.

- The achievement of any one will have an affect on the achievement of others.
- Sometimes the effect is positive and sometimes negative.



# Add WeChat powcoder

# Table of Contents

- ① Course Introduction
- ② About Me
- ③ Introduction to Software Architecture

# Assignment Project Exam Help

Why Is Architecture Important?

What Makes a Good Architect?

- ④ Practicing Software Architecture

Foundations

Structures and Views [Bass et al.]

Aspects of Software Architecture [Vogel et al.]

4+1 Architecture View Model [Kruchten]

C4 Model [Brown]

# Add WeChat powcoder

- ⑤ Introduction to Quality Attributes

Architecturally Significant Requirements

Quality Attribute Scenarios

Architectural Tactics

- ⑥ Wrap-Up

# Quality Attribute Scenarios

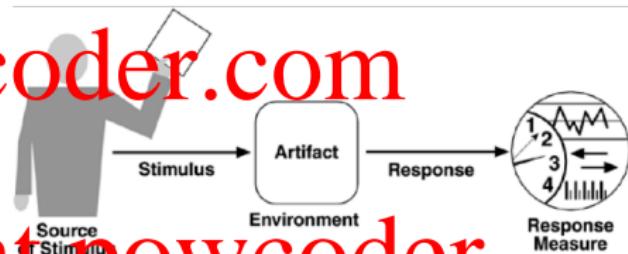
- *Stimulus*: An event arriving at the system

- *Stimulus source*: Where the stimulus came from

- *Environment*: The set of circumstances in which the scenario takes place

- *Artifact*: The portion of the system to which the requirement arrives

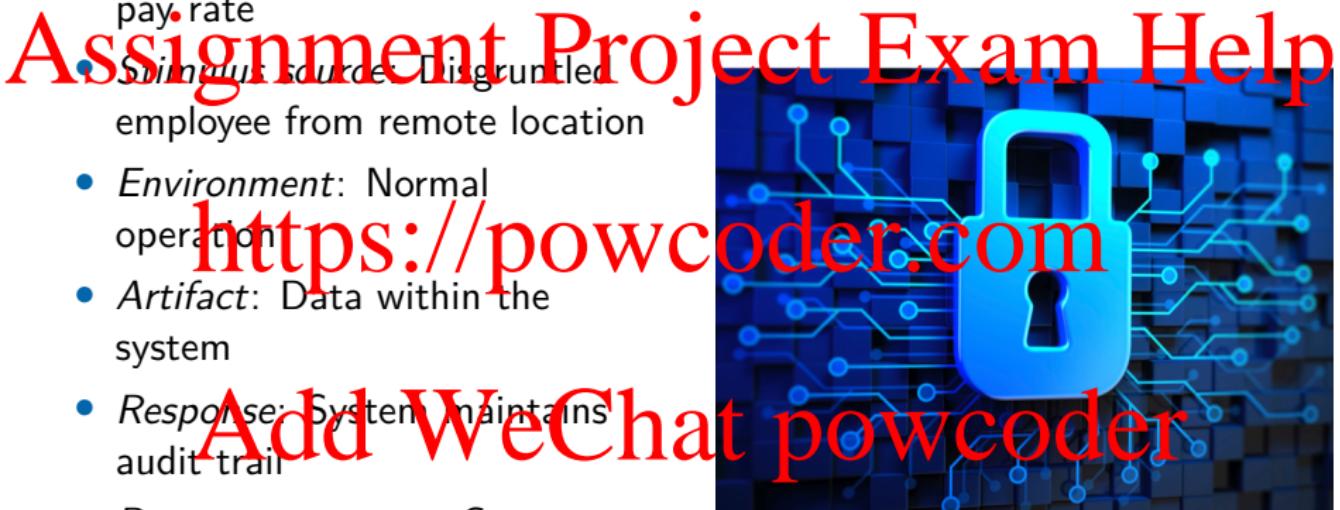
- *Response*: How the system should respond to the stimulus
- *Response measure*: How to evaluate whether the response is satisfactory



Add WeChat powcoder

## Example Scenario (Security)

- *Stimulus*: Attempts to modify pay rate
- *Stimulus source*: Disgruntled employee from remote location
- *Environment*: Normal operation
- *Artifact*: Data within the system
- *Response*: System maintains audit trail
- *Response measure*: Correct data is restored within a day and source of tampering identified



## Table of Contents

- 1 Course Introduction
- 2 About Me
- 3 Introduction to Software Architecture

# Assignment Project Exam Help

Why Is Architecture Important?

What Makes a Good Architect?

- 4 Practicing Software Architecture

Foundations

Structures and Views [Bass et al.]

Aspects of Software Architecture [Vogel et al.]

4+1 Architecture View Model [Kruchten]

C4 Model [Brown]

# Add WeChat powcoder

- 5 Introduction to Quality Attributes

Architecturally Significant Requirements

Quality Attribute Scenarios

Architectural Tactics

- 6 Wrap-Up

# Assignment Project Exam Help

- An *architectural tactic* is a design decision that influences the achievement of a quality attribute response
- In other words, tactics directly affect the system's response to some stimulus
- A tactic is focused on a single quality attribute response. Within a tactic, there is no consideration of tradeoffs.
- A set of tactics is an *architectural strategy*

Add WeChat powcoder

# Table of Contents

- ① Course Introduction
- ② About Me
- ③ Introduction to Software Architecture

# Assignment Project Exam Help

Why Is Architecture Important?

What Makes a Good Architect?

- ④ Practicing Software Architecture

Foundations

Structures and Views [Bass et al.]

Aspects of Software Architecture [Vogel et al.]

4+1 Architecture View Model [Kruchten]

C4 Model [Brown]

# Add WeChat powcoder

- ⑤ Introduction to Quality Attributes

Architecturally Significant Requirements

Quality Attribute Scenarios

Architectural Tactics

- ⑥ Wrap-Up

# Assignment Project Exam Help

- Read chapters 1-4 from SAIP
- An attendance survey is now available on D2L. It is due Thursday, September 17 at 5:30PM.
- Homework 1 is now available on D2L. It is due Thursday, September 24 at 5:30PM.
- Quiz 1 will be available next week (it includes questions from week 2's lecture). It will also be due Thursday, September 24 at 5:30PM.

<https://powcoder.com>  
Add WeChat powcoder

# Assignment Project Exam Help

- Topics: Availability, safety
- <https://powcoder.com>

Add WeChat powcoder

# Assignment Project Exam Help

- <https://powcoder.com>  
Add WeChat powcoder
- [BCK12] Len Bass, Paul Clements, and Rick Kazman. *Software Architecture in Practice*. Addison-Wesley Professional, 3rd edition, 2012.
  - [Fow02] Martin Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.
  - [Bos04] Jan Bosch. Software architecture: The next step. In Flavio Duquennoy, Léan C. Wariooy, and Tony Morrison, Editors, *Software Architecture*, pages 194–199, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
  - [SG09] Diomidis Spinellis and Georgios Gousios. *Beautiful Architecture: Leading Thinkers Reveal the Hidden Beauty in Software Design*. O'Reilly Media, Inc., 1st edition, 2009.
  - [VACK11] Oliver Vogel, Ingo Arnold, Arif Chughtai, and Timo Kehrer. *Software Architecture - A Comprehensive Framework and Guide for Practitioners*. Springer, 2011.
  - [Bro19] Simon Brown. The c4 model for visualizing software architecture, 2019.
  - [Fou18] The Apache Software Foundation. Hdfs architecture, 2018.