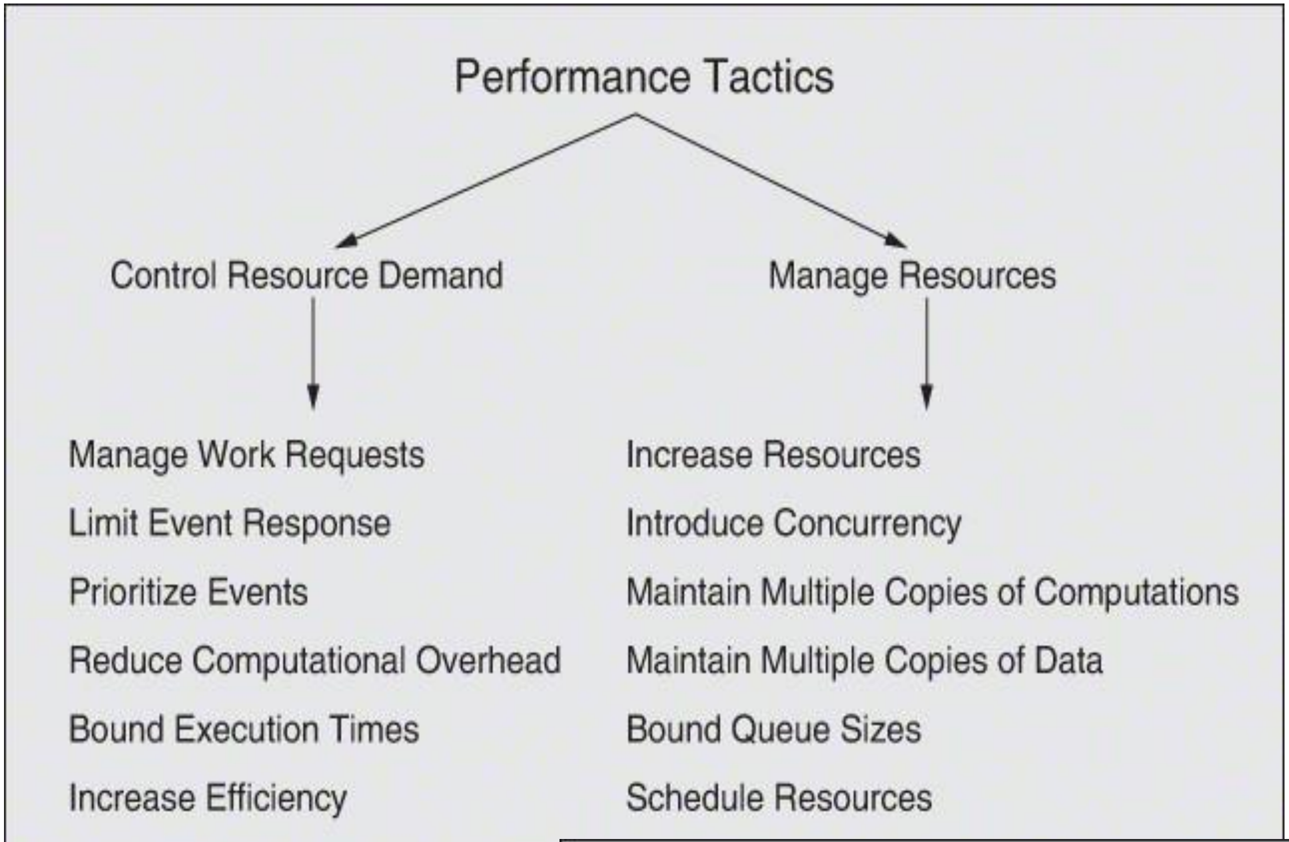
SENG 350
- Software Architecture & Design

Shuja Mughal

Architectural Tactics

Fall 2024

University
of Victoria



Performance Tactics

Control Resource Demand

Manage Resources

Manage Work Requests

Limit Event Response

Prioritize Events

Reduce Computational Overhead

Bound Execution Times

Increase Efficiency

Increase Resources

Introduce Concurrency

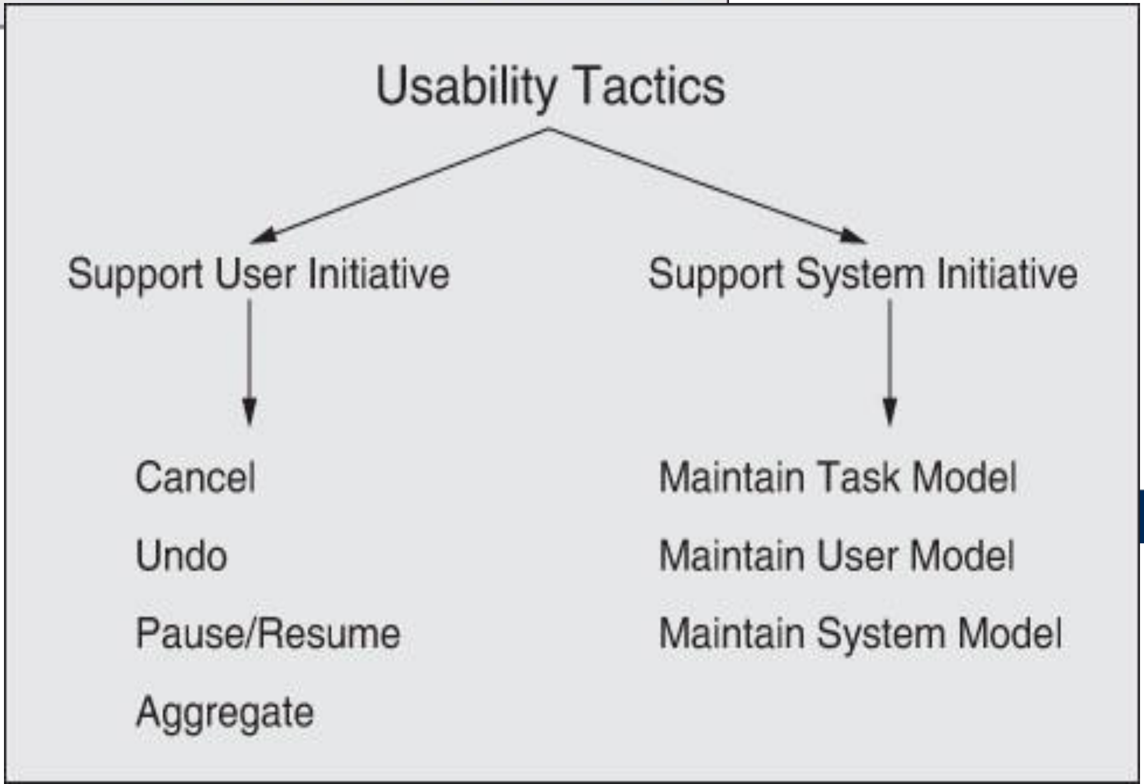
Maintain Multiple Copies of Computations

Maintain Multiple Copies of Data

Bound Queue Sizes

Schedule Resources

Last Time



Usability Tactics

Support User Initiative

Support System Initiative

Cancel

Undo

Pause/Resume

Aggregate

Maintain Task Model

Maintain User Model

Maintain System Model

Milestone 3

1. Construct incrementally

- This milestone is the second of three major design and construction "sprints".
- Your system is not expected to be complete at the end of milestone 3, but defined parts of the system should be complete and deployable.
- Make sure you deliberately plan the parts you want to finish in this "sprint".

University of Victoria

Milestone 3

2. Architectural Tactic for Availability

- Work through the checklist to decide how to address the quality attribute of availability in your system.
- Select appropriate architectural tactics to ensure availability based on your decisions.
- Your architecture should use at least five architectural availability tactics.
- Implement the code for these five architectural tactics in your project

University of Victoria

Milestone 3

3. Design Patterns

- Identify and plan five design patterns in your design and system.
- It should be five patterns that you implement.

Milestone 3

4. Design Diagrams and System Description

- Update all the previous UML Diagrams along with the System Description according to feasibility and changed requirements.
- This should be in detail as to why they were changed and why they were not feasible.

Milestone 3

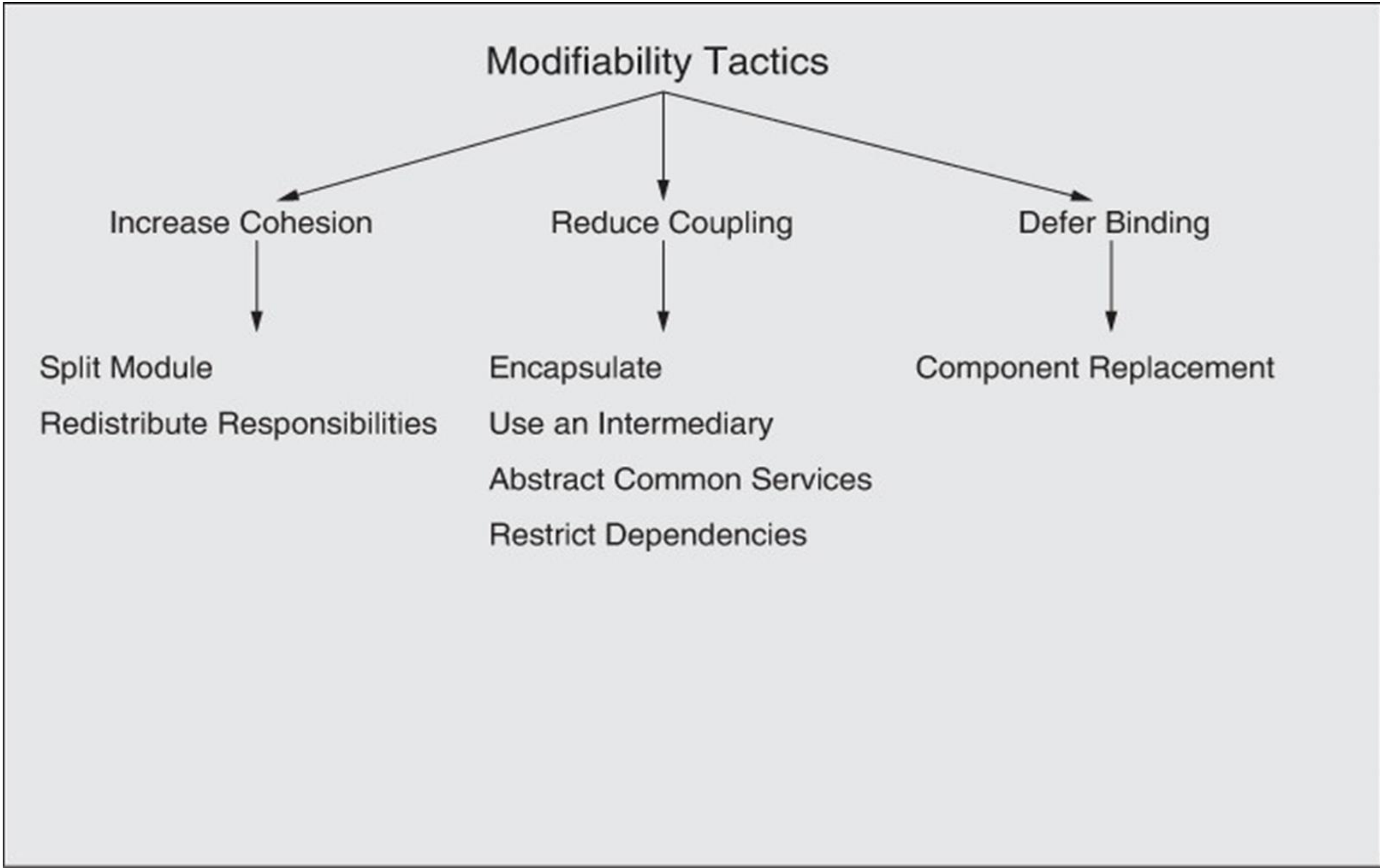
5. Contributions

- Summarize the contributions made by each team member.
- Point out the sections/parts each group member was "most responsible" for.

Modifiability

University
of Victoria

Modifiability



Modifiability Tactics

Increase Cohesion

- Split Module
- Redistribute Responsibilities

Reduce Coupling

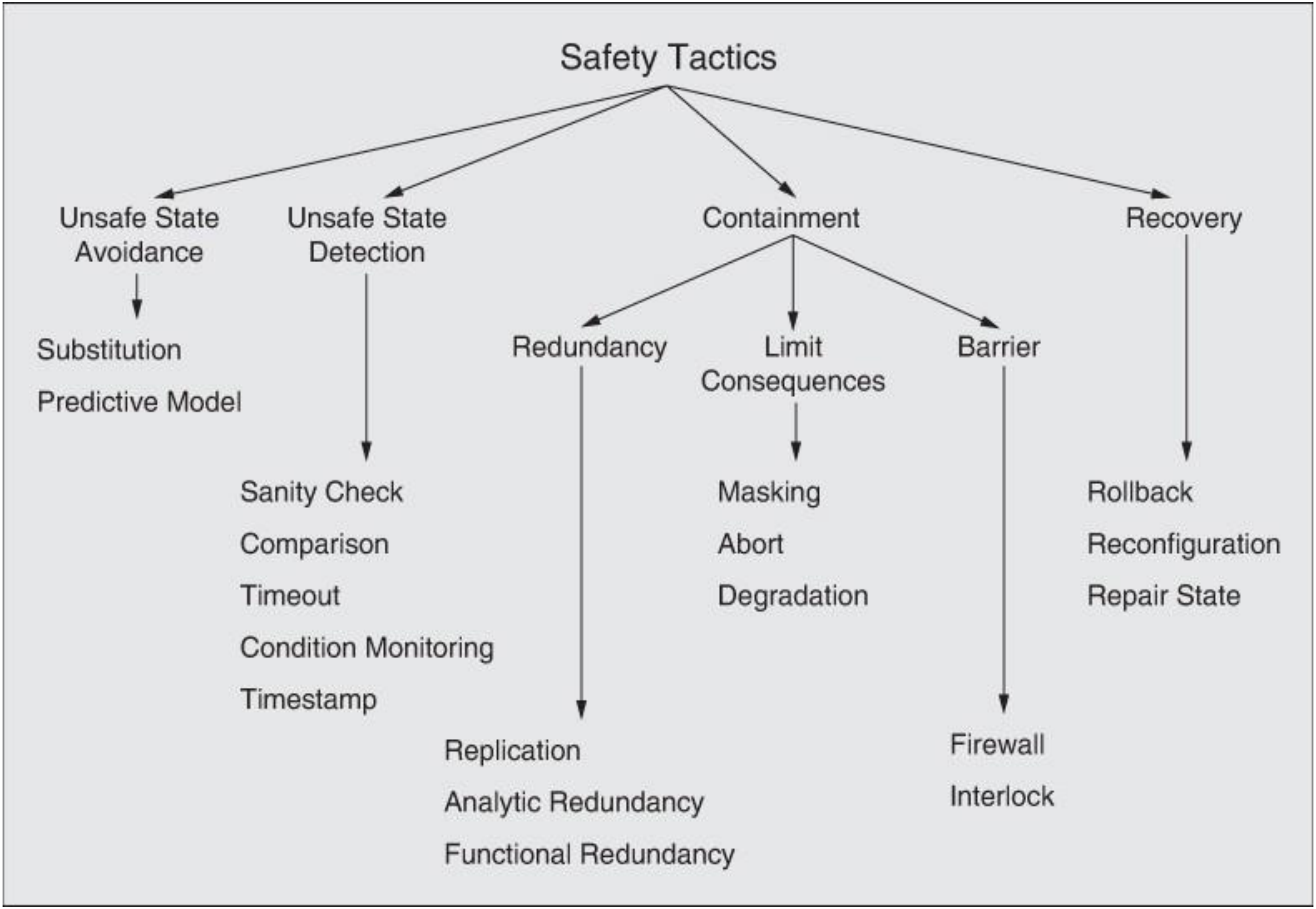
- Encapsulate
- Use an Intermediary
- Abstract Common Services
- Restrict Dependencies

Defer Binding

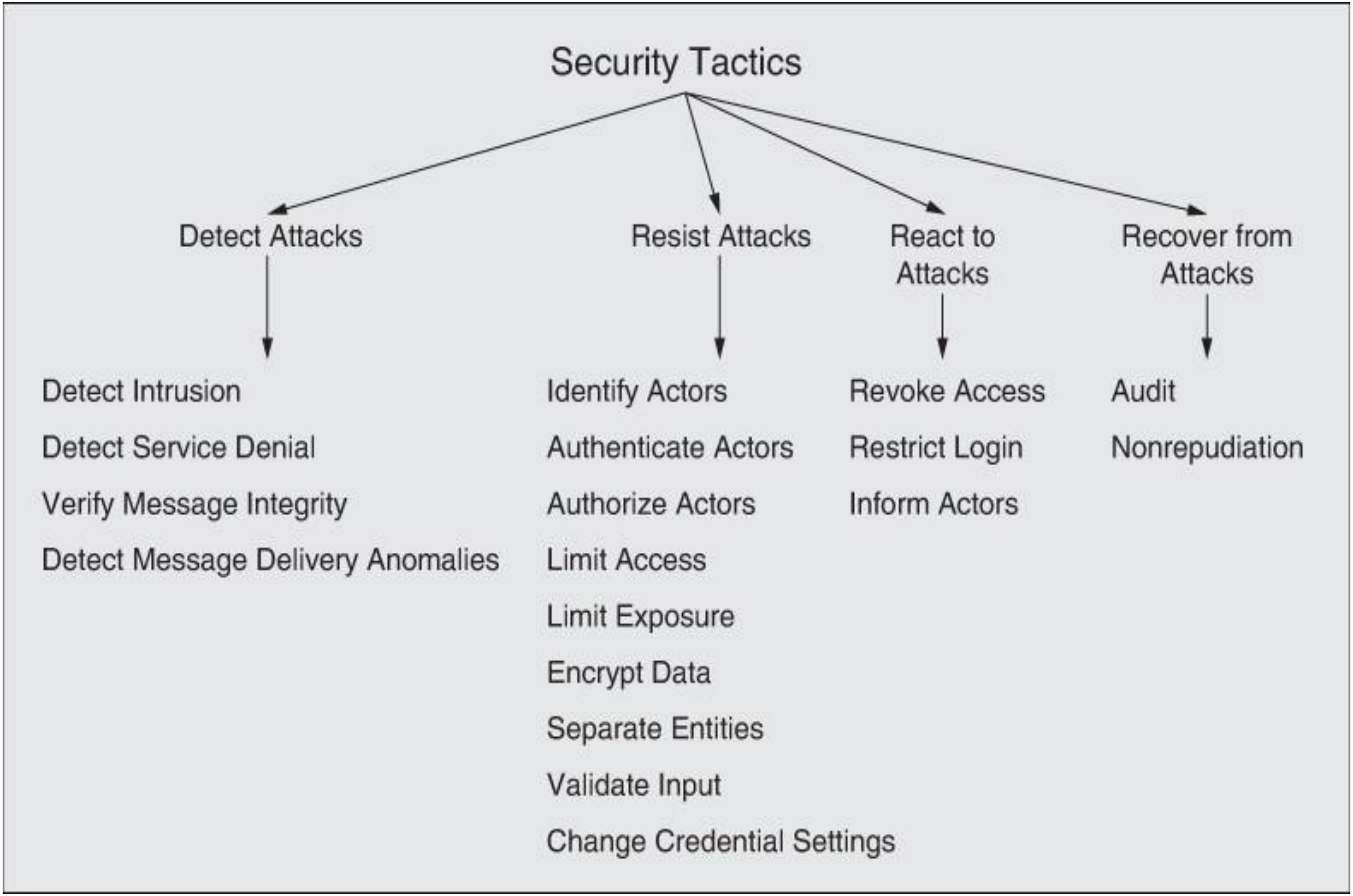
- Component Replacement

Safety

University
of Victoria

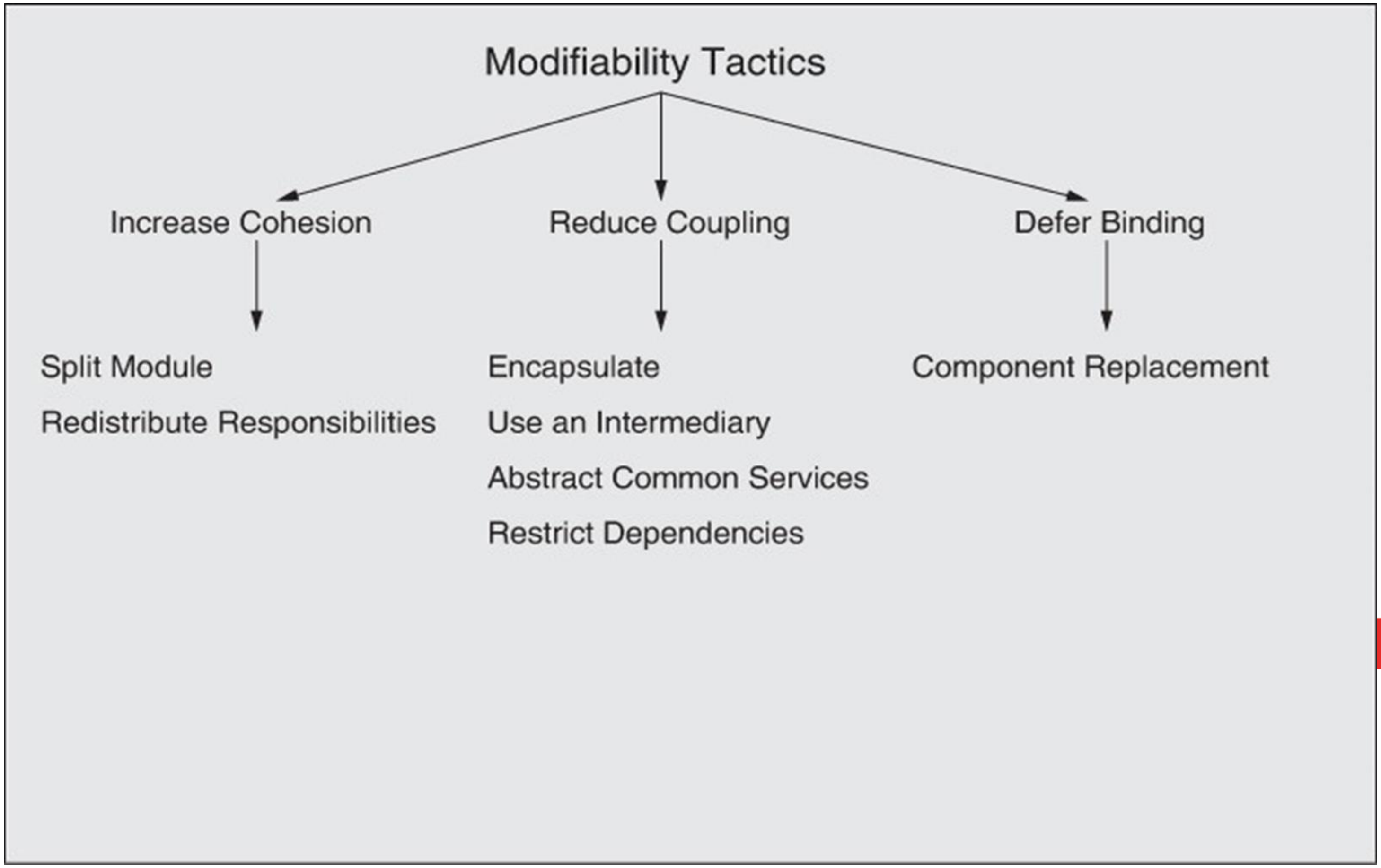

Safety Tactics
Unsafe State Avoidance
Substitution
Predictive Model
Unsafe State Detection
Sanity Check
Comparison
Timeout
Condition Monitoring
Timestamp
Containment
Redundancy
Replication
Analytic Redundancy
Functional Redundancy
Limit Consequences
Masking
Abort
Degradation
Barrier
Firewall
Interlock
Recovery
Rollback
Reconfiguration
Repair State

Security


Security Tactics
Detect Attacks
Resist Attacks
React to Attacks
Recover from Attacks
Detect Intrusion
Detect Service Denial
Verify Message Integrity
Detect Message Delivery Anomalies
Identify Actors
Authenticate Actors
Authorize Actors
Limit Access
Limit Exposure
Encrypt Data
Separate Entities
Validate Input
Change Credential Settings
Revoke Access
Restrict Login
Inform Actors
Audit
Nonrepudiation

Activity

- Make a group of 2.
- Pick an example with multiple functions.
- Implement 2 of the modifiability tactics.
- One from Increase Cohesion and one from Reduce Coupling.
- Submit screenshots of the code and the outputs to teams.


Modifiability Tactics
Increase Cohesion
Reduce Coupling
Defer Binding
Split Module
Redistribute Responsibilities
Encapsulate
Use an Intermediary
Abstract Common Services
Restrict Dependencies
Component Replacement