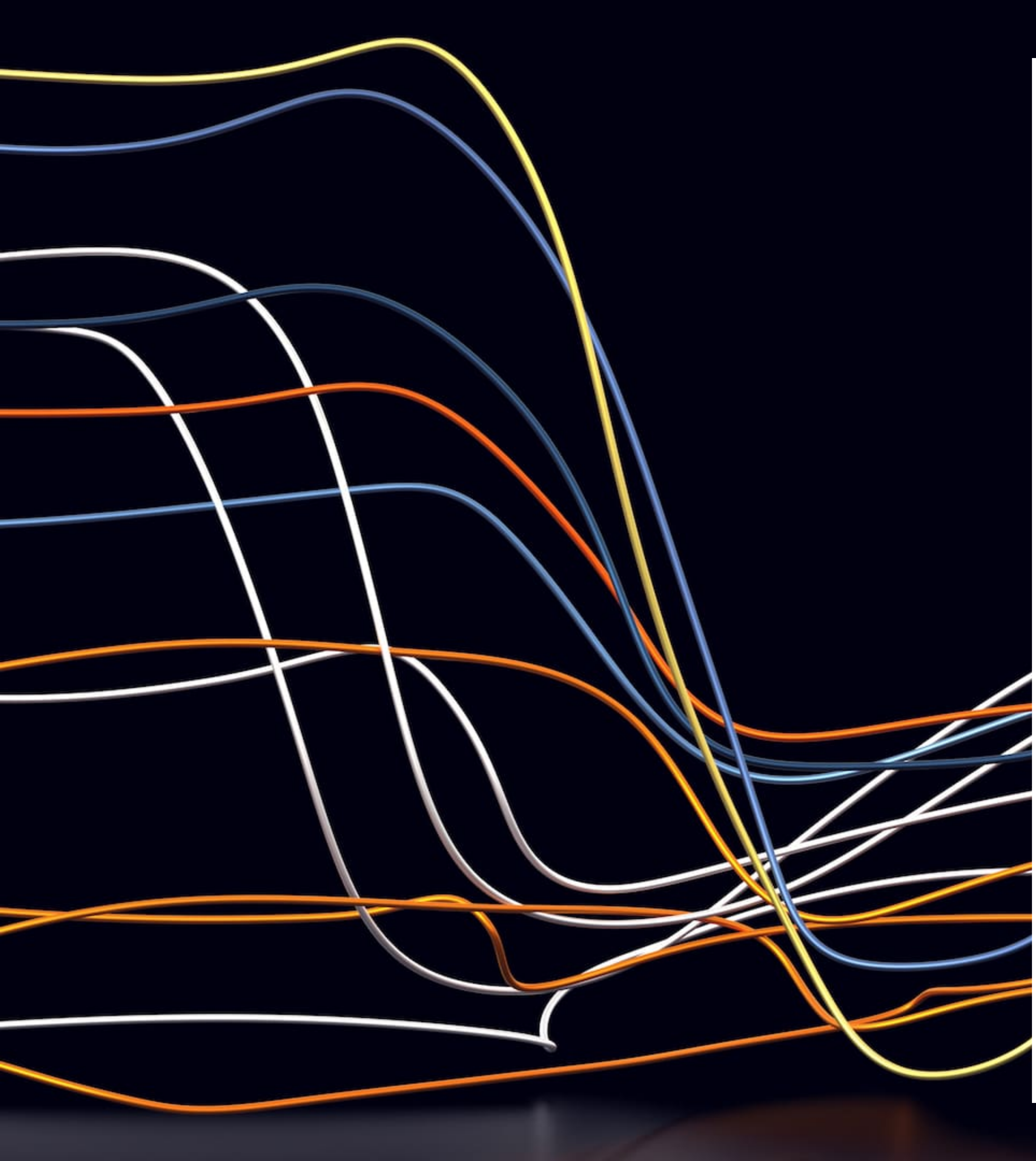


Cohesion

in Modeling and Design

INNOQ

MICHAEL PLÖD

FELLOW

Michael Plöd

Fellow at INNOQ

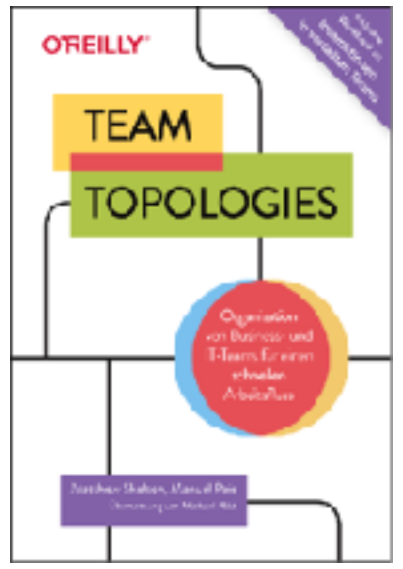
Socials: **@bitboss** (@mastodon.social / .bsky.social)

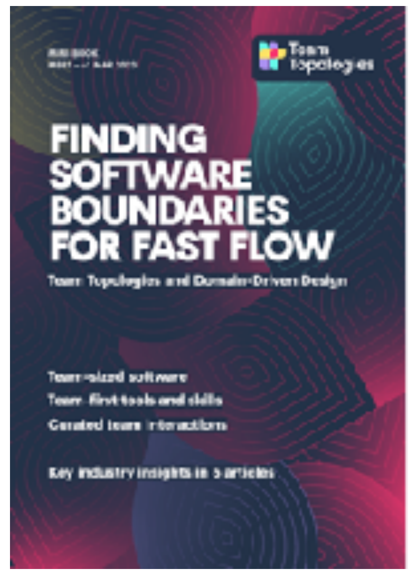
LinkedIn: https://www.linkedin.com/in/michael-ploed/

Current consulting topics:

- Domain-Driven Design
- Team Topologies
- Transformation from IT Delivery to digital product orgs

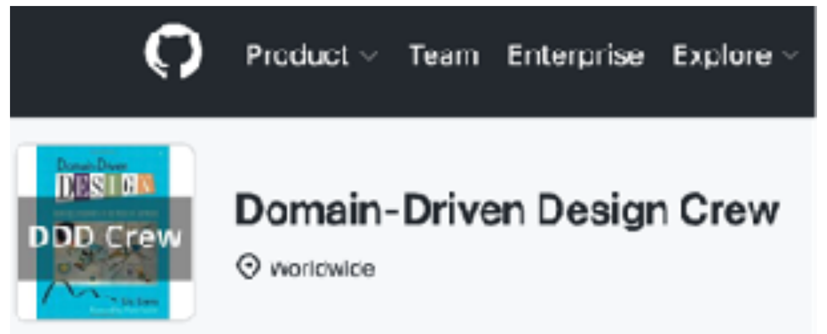
Regular speaker at (inter-)national conferences and author of books + various articles




FINDING SOFTWARE BOUNDARIES FOR FAST FLOW
Team Topologies and Domain-Driven Design

Team Topologies

Advocate



Domain-Driven Design Crew

worldwide

Modularity

Let’s revisit some basics first

Separation of Concerns

is the division of complex systems according to responsibility

Modularity

is a specialization of

Separation Of Concerns

Modularity

is a specialization of
Separation Of Concerns

+ Information Hiding

+ Loose Coupling

+ High Cohesion

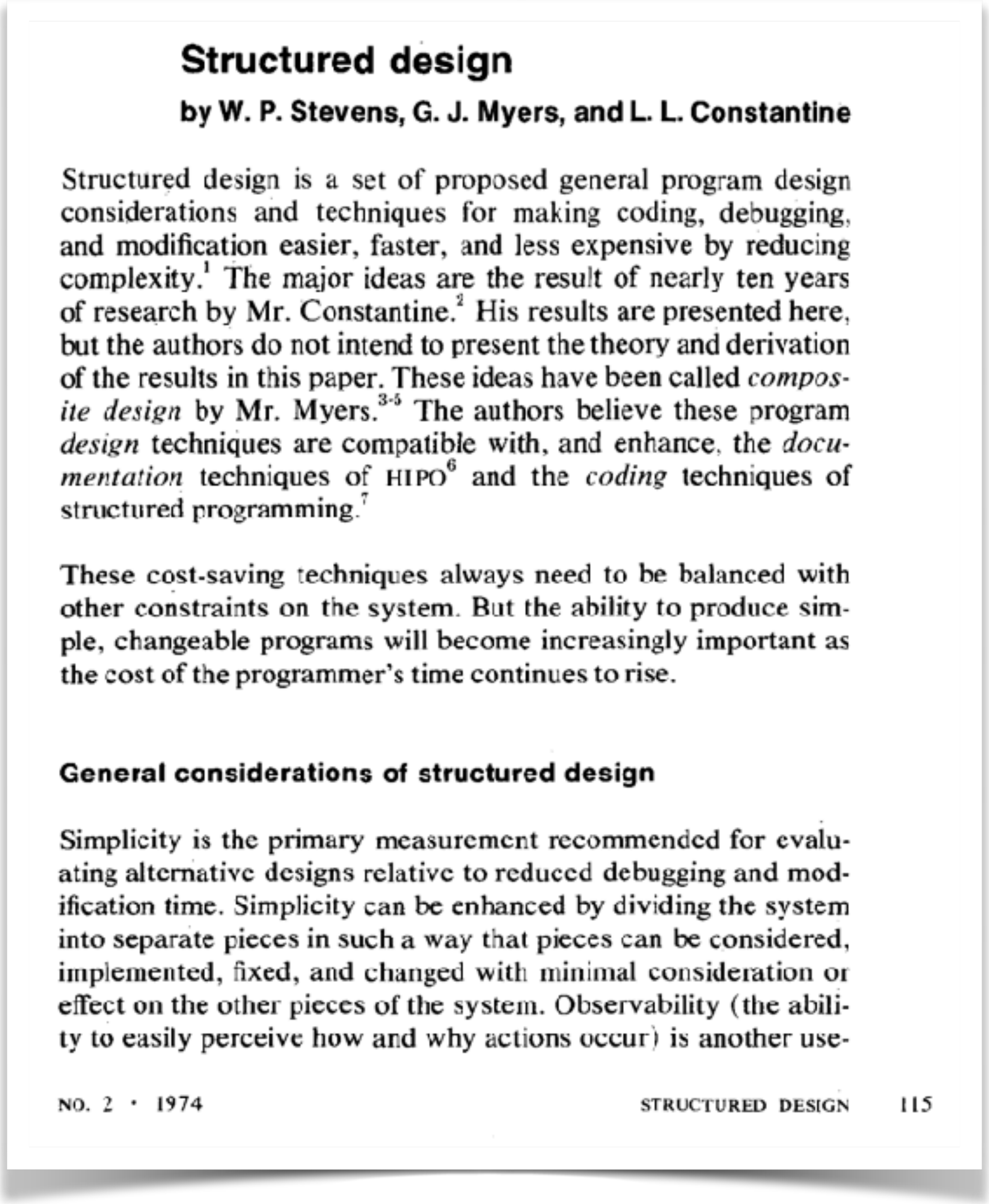
We talk a lot about coupling

„Coupling is reduced when the relationships among elements not in the same module are minimized. There are two ways of achieving this

minimizing the relationships among modules

and

maximizing relationships among elements in the same module“



Structured design

by W. P. Stevens, G. J. Myers, and L. L. Constantine

Structured design is a set of proposed general program design considerations and techniques for making coding, debugging, and modification easier, faster, and less expensive by reducing complexity.[1] The major ideas are the result of nearly ten years of research by Mr. Constantine.[2] His results are presented here, but the authors do not intend to present the theory and derivation of the results in this paper. These ideas have been called *composite design* by Mr. Myers.[3-5] The authors believe these program *design* techniques are compatible with, and enhance, the *documentation* techniques of HIPO[6] and the *coding* techniques of structured programming.[7]

These cost-saving techniques always need to be balanced with other constraints on the system. But the ability to produce simple, changeable programs will become increasingly important as the cost of the programmer's time continues to rise.

General considerations of structured design

Simplicity is the primary measurement recommended for evaluating alternative designs relative to reduced debugging and modification time. Simplicity can be enhanced by dividing the system into separate pieces in such a way that pieces can be considered, implemented, fixed, and changed with minimal consideration or effect on the other pieces of the system. Observability (the ability to easily perceive how and why actions occur) is another use-

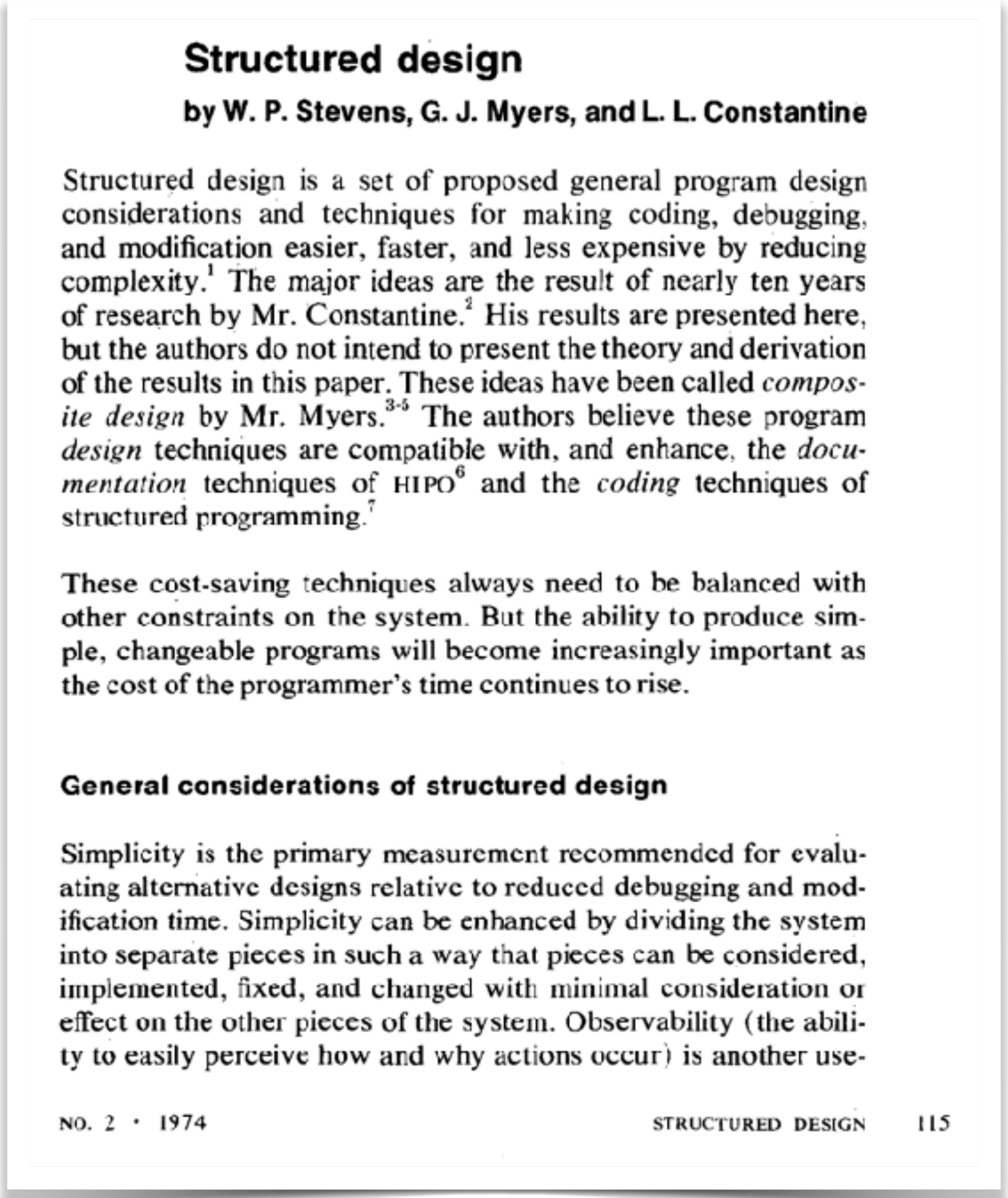
Wayne P. Stevens, Glenford J. Myers, Larry L. Constantine, **Structured Design**, IBM Systems Journal, 13(2):115-139, 1974.

„Coupling is reduced when the relationships among elements not in the same module are minimized. There are two ways of achieving this

minimizing the relationships among modules

and

maximizing relationships among elements in the same module"



Structured design

by W. P. Stevens, G. J. Myers, and L. L. Constantine

Structured design is a set of proposed general program design considerations and techniques for making coding, debugging, and modification easier, faster, and less expensive by reducing complexity.[1] The major ideas are the result of nearly ten years of research by Mr. Constantine.[2] His results are presented here, but the authors do not intend to present the theory and derivation of the results in this paper. These ideas have been called *composite design* by Mr. Myers.[3-5] The authors believe these program *design* techniques are compatible with, and enhance, the *documentation* techniques of HIPO[6] and the *coding* techniques of structured programming.[7]

These cost-saving techniques always need to be balanced with other constraints on the system. But the ability to produce simple, changeable programs will become increasingly important as the cost of the programmer's time continues to rise.

General considerations of structured design

Simplicity is the primary measurement recommended for evaluating alternative designs relative to reduced debugging and modification time. Simplicity can be enhanced by dividing the system into separate pieces in such a way that pieces can be considered, implemented, fixed, and changed with minimal consideration or effect on the other pieces of the system. Observability (the ability to easily perceive how and why actions occur) is another use-

NO. 2 · 1974 STRUCTURED DESIGN 115

Wayne P. Stevens, Glenford J. Myers, Larry L. Constantine, **Structured Design**, IBM Systems Journal, 13(2):115-139, 1974.

We talk a lot about coupling...

... but not so much about cohesion

COHERERE

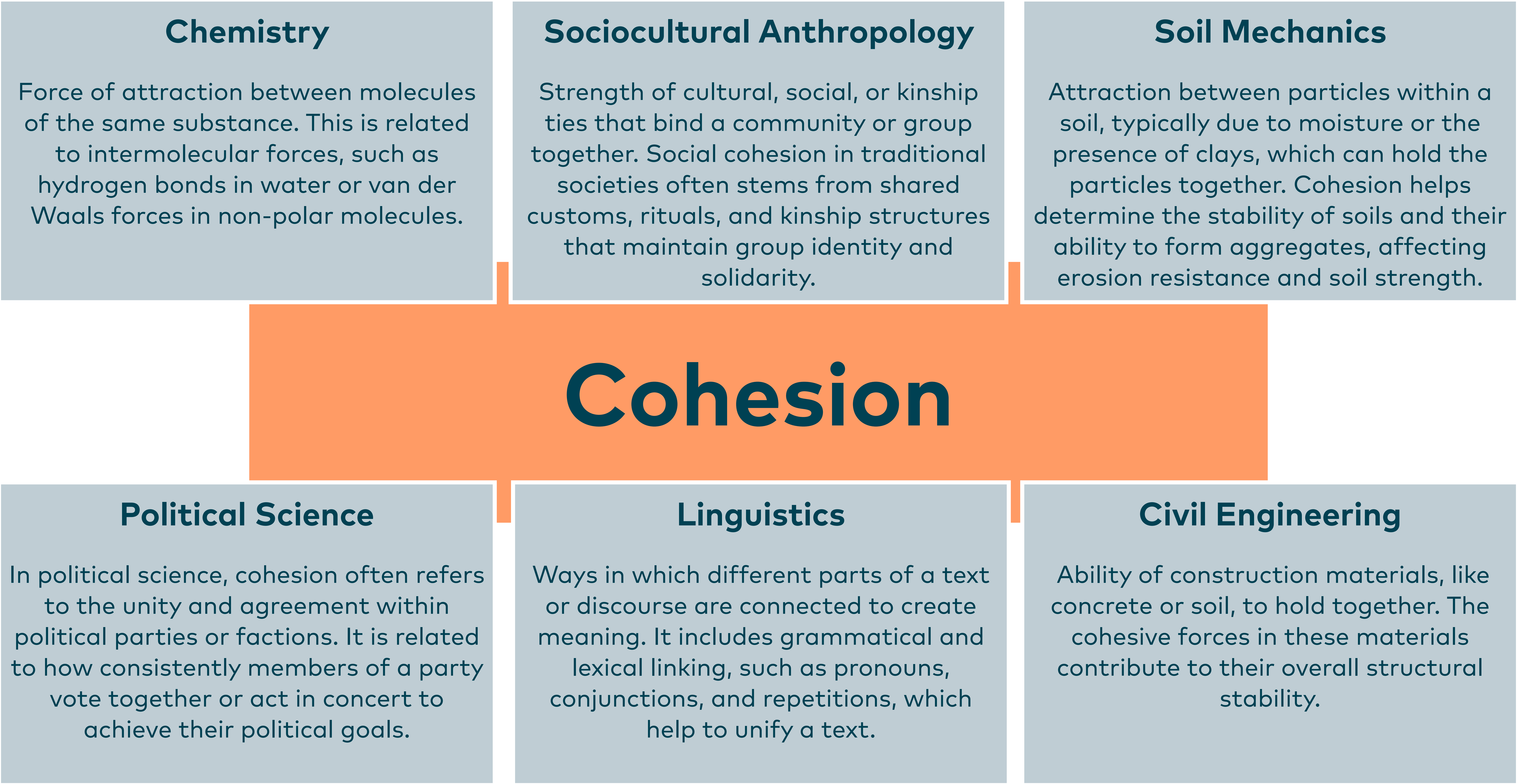
Latin verb

cohereo,
coheres,
coherere,
cohesi,
cohesum

- to stick/cling/hold/grow together
- to adhere
- to embrace
- to touch
- to adjoin
- to be in contact

Cohesion

Is not just an IT thing



Cohesion

Chemistry

Force of attraction between molecules of the same substance. This is related to intermolecular forces, such as hydrogen bonds in water or van der Waals forces in non-polar molecules.

Sociocultural Anthropology

Strength of cultural, social, or kinship ties that bind a community or group together. Social cohesion in traditional societies often stems from shared customs, rituals, and kinship structures that maintain group identity and solidarity.

Soil Mechanics

Attraction between particles within a soil, typically due to moisture or the presence of clays, which can hold the particles together. Cohesion helps determine the stability of soils and their ability to form aggregates, affecting erosion resistance and soil strength.

Political Science

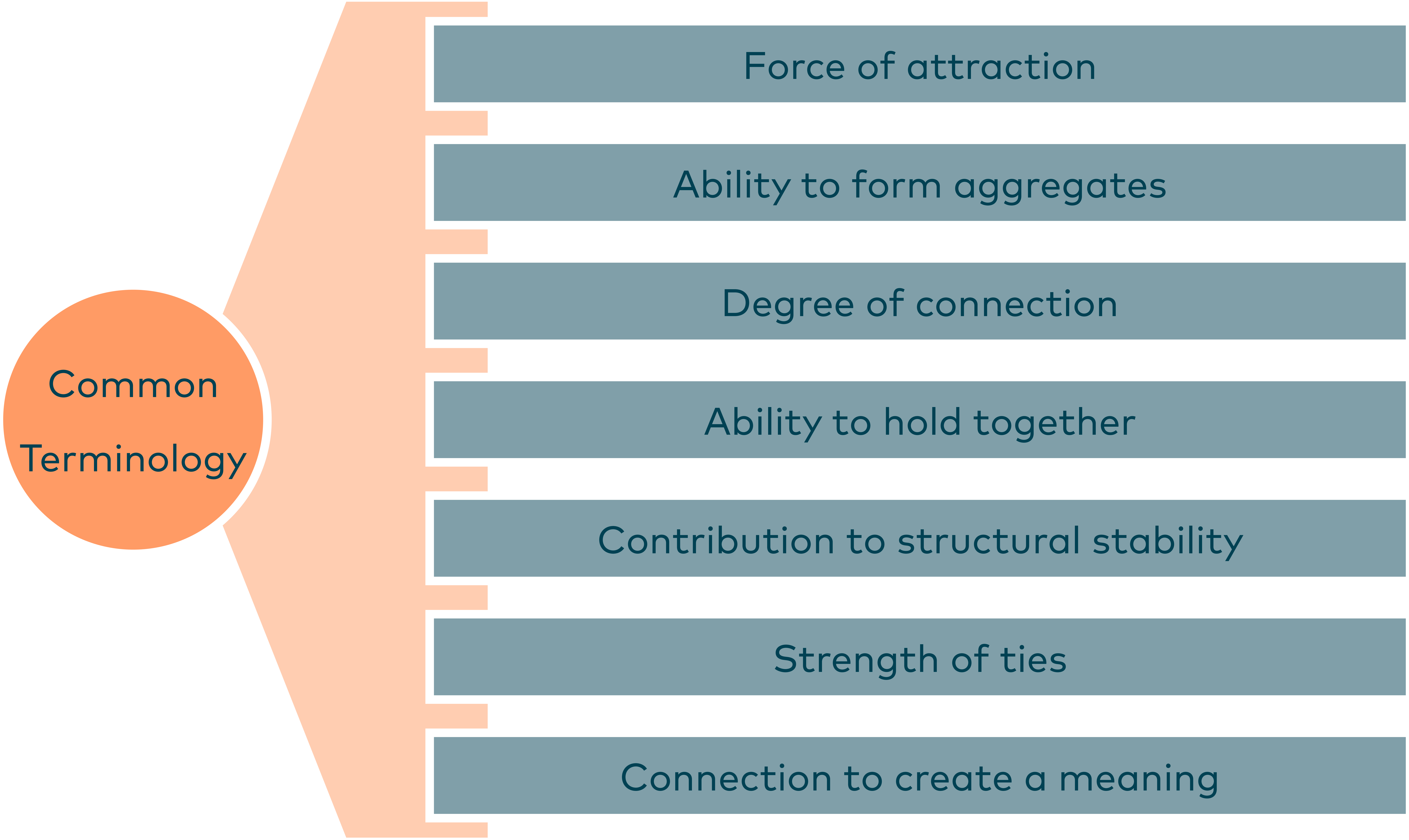
In political science, cohesion often refers to the unity and agreement within political parties or factions. It is related to how consistently members of a party vote together or act in concert to achieve their political goals.

Linguistics

Ways in which different parts of a text or discourse are connected to create meaning. It includes grammatical and lexical linking, such as pronouns, conjunctions, and repetitions, which help to unify a text.

Civil Engineering

Ability of construction materials, like concrete or soil, to hold together. The cohesive forces in these materials contribute to their overall structural stability.



Common Terminology

- Force of attraction
- Ability to form aggregates
- Degree of connection
- Ability to hold together
- Contribution to structural stability
- Strength of ties
- Connection to create a meaning

Which of these perspectives are relevant for cohesion in IT?

Sociocultural Anthropology

Strength of cultural, social, or kinship ties that bind a community or group together. Social cohesion in traditional societies often stems from shared customs, rituals, and kinship structures that maintain group identity and solidarity.

Soil Mechanics

Attraction between particles within a soil, typically due to moisture or the presence of clays, which can hold the particles together. Cohesion helps determine the stability of soils and their ability to form aggregates, affecting erosion resistance and soil strength.

Chemistry

Force of attraction between molecules of the same substance. This is related to intermolecular forces, such as hydrogen bonds in water or van der Waals forces in non-polar molecules.

Civil Engineering

Ability of construction materials, like concrete or soil, to hold together. The cohesive forces in these materials contribute to their overall structural stability.

Political Science

In political science, cohesion often refers to the unity and agreement within political parties or factions. It is related to how consistently members of a party vote together or act in concert to achieve their political goals.

Linguistics

Ways in which different parts of a text or discourse are connected to create meaning. It includes grammatical and lexical linking, such as pronouns, conjunctions, and repetitions, which help to unify a text.

ALL

Software Architecture: Structural Cohesion

Soil Mechanics

Attraction between particles within a soil, typically due to moisture or the presence of clays, which can hold the particles together. Cohesion helps determine the stability of soils and their ability to form aggregates, affecting erosion resistance and soil strength.

Chemistry

Force of attraction between molecules of the same substance. This is related to intermolecular forces, such as hydrogen bonds in water or van der Waals forces in non-polar molecules.

Civil Engineering

Ability of construction materials, like concrete or soil, to hold together. The cohesive forces in these materials contribute to their overall structural stability.

Work in organization:

Social / linguistic Cohesion

Sociocultural Anthropology

Strength of cultural, social, or kinship ties that bind a community or group together. Social cohesion in traditional societies often stems from shared customs, rituals, and kinship structures that maintain group identity and solidarity.

Political Science

In political science, cohesion often refers to the unity and agreement within political parties or factions. It is related to how consistently members of a party vote together or act in concert to achieve their political goals.

Linguistics

Ways in which different parts of a text or discourse are connected to create meaning. It includes grammatical and lexical linking, such as pronouns, conjunctions, and repetitions, which help to unify a text.

Software Architecture:

Structural Cohesion

Work in organization:

Social / linguistic Cohesion

Sociocultural Anthropology

Strength of cultural, social, or kinship ties that bind a community or group together. Social cohesion in traditional societies often stems from shared customs, rituals, and kinship structures that maintain group identity and solidarity.

Soil Mechanics

Attraction between particles within a soil, typically due to moisture or the presence of clays, which can hold the particles together. Cohesion helps determine the stability of soils and their ability to form aggregates, affecting erosion resistance and soil strength.

Chemistry

Force of attraction between molecules of the same substance. This is related to intermolecular forces, such as hydrogen bonds in water or van der Waals forces in non-polar molecules.

Civil Engineering

Ability of construction materials, like concrete or soil, to hold together. The cohesive forces in these materials contribute to their overall structural stability.

Political Science

In political science, cohesion often refers to the unity and agreement within political parties or factions. It is related to how consistently members of a party vote together or act in concert to achieve their political goals.

Linguistics

Ways in which different parts of a text or discourse are connected to create meaning. It includes grammatical and lexical linking, such as pronouns, conjunctions, and repetitions, which help to unify a text.

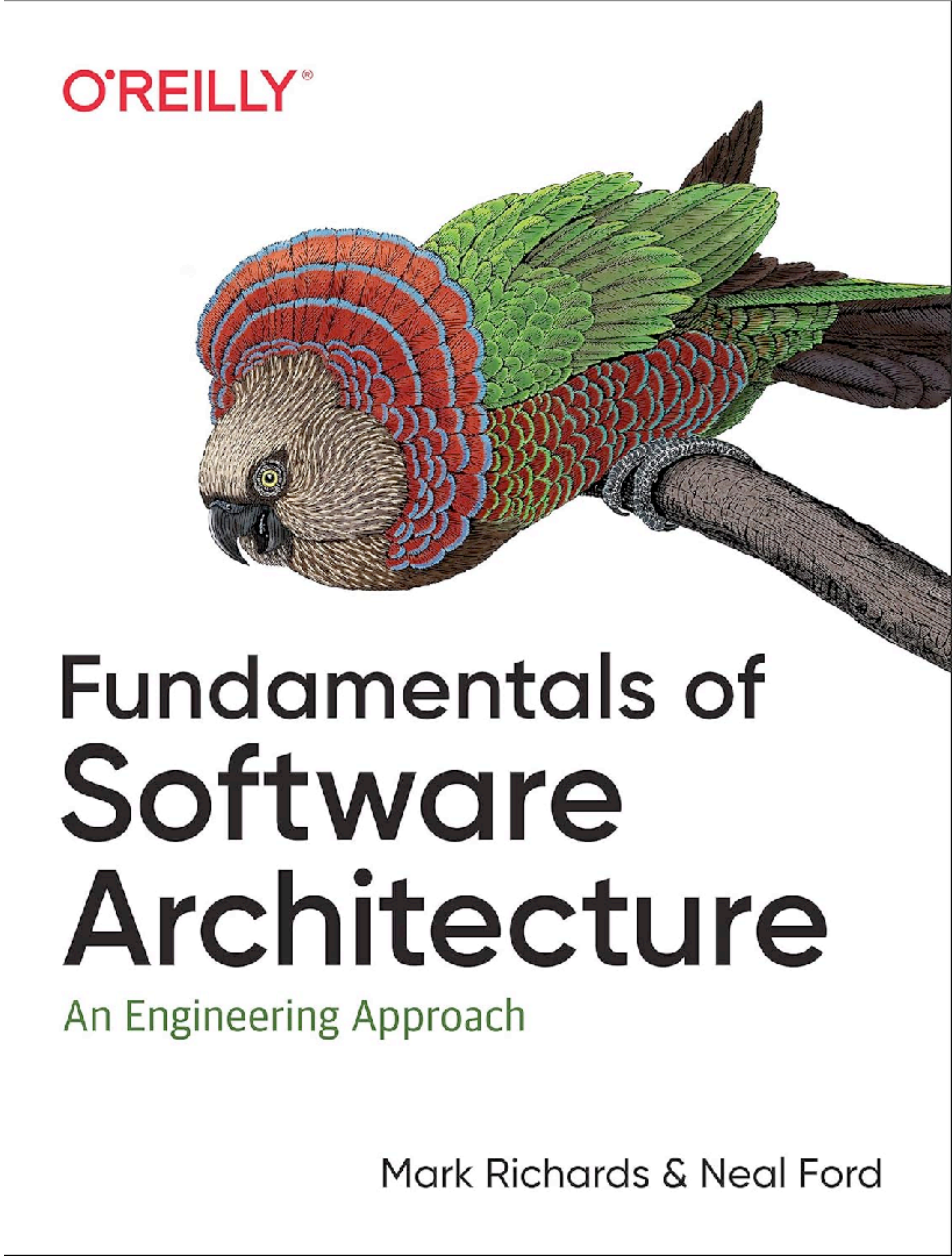
Cohesion

in

Software Architecture

Cohesion refers to what extent the parts of a module should be contained within the same module.

In other words, it is a measure of how related the parts are to one another.



O'REILLY®

Fundamentals of Software Architecture

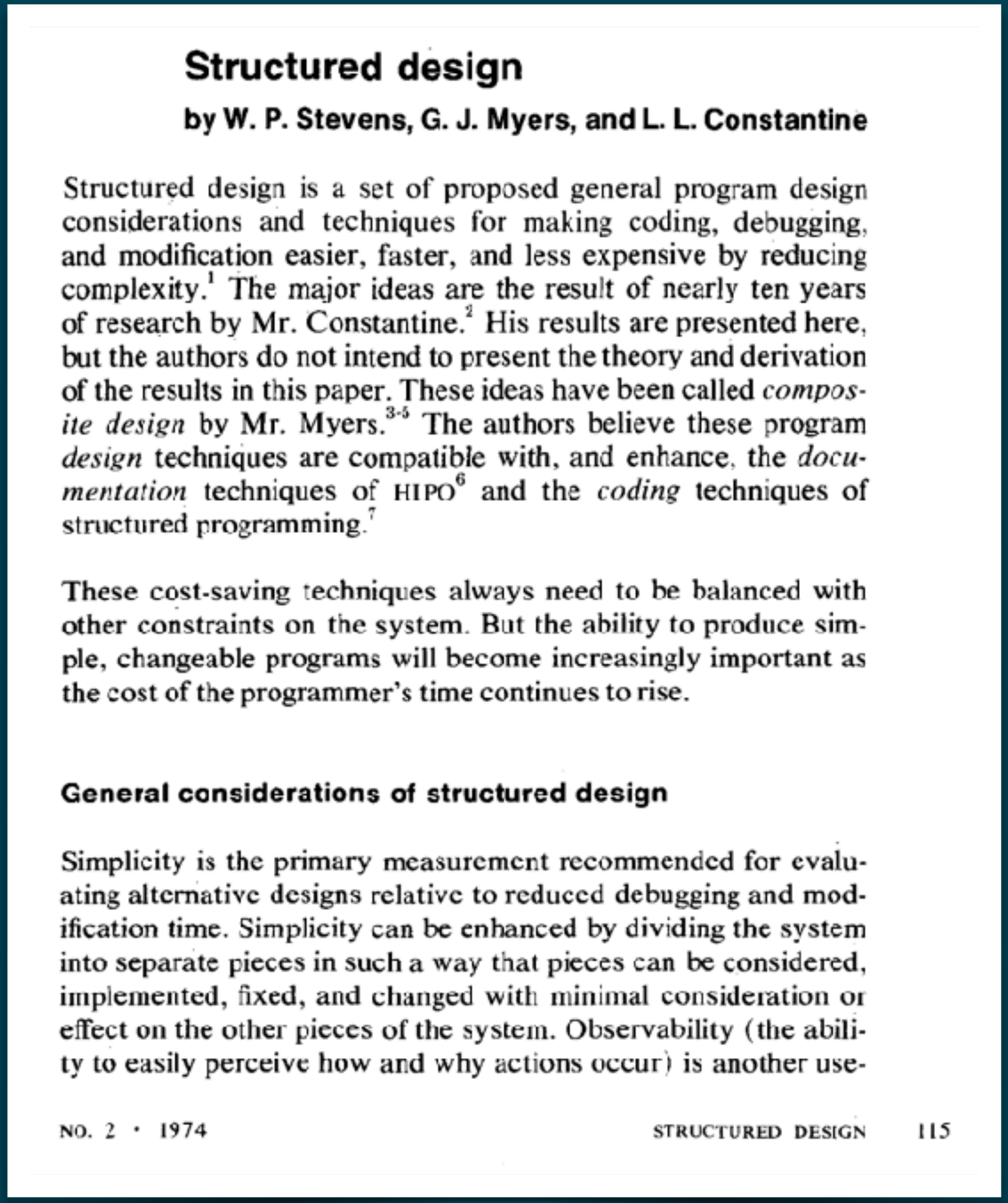
An Engineering Approach

Mark Richards & Neal Ford

7 Levels of SMC Cohesion

SMC stands for the authors of the Structured Design paper:

Stevens, **M**yers, **C**onstantine



Structured design

by W. P. Stevens, G. J. Myers, and L. L. Constantine

Structured design is a set of proposed general program design considerations and techniques for making coding, debugging, and modification easier, faster, and less expensive by reducing complexity.[1] The major ideas are the result of nearly ten years of research by Mr. Constantine.[2] His results are presented here, but the authors do not intend to present the theory and derivation of the results in this paper. These ideas have been called *composite design* by Mr. Myers.[3-5] The authors believe these program *design* techniques are compatible with, and enhance, the *documentation* techniques of HIPO[6] and the *coding* techniques of structured programming.[7]

These cost-saving techniques always need to be balanced with other constraints on the system. But the ability to produce simple, changeable programs will become increasingly important as the cost of the programmer's time continues to rise.

General considerations of structured design

Simplicity is the primary measurement recommended for evaluating alternative designs relative to reduced debugging and modification time. Simplicity can be enhanced by dividing the system into separate pieces in such a way that pieces can be considered, implemented, fixed, and changed with minimal consideration or effect on the other pieces of the system. Observability (the ability to easily perceive how and why actions occur) is another use-

7 Levels of **SMC Cohesion**

Functional Cohesion

The elements of the module are grouped because they all contribute to performing a single, well-defined task. Every part of the module is essential to its overall functionality.

Sequential Cohesion

The elements of the module are grouped because the output from one element serves as input to another. This type of cohesion exists when functions are interdependent in a linear chain.

Communicational Cohesion

The elements of the module are grouped because they operate on the same data or contribute to the same data structure.

Procedural Cohesion*

The elements of the module are grouped because they always follow a specific sequence of execution, even if they are not closely related by purpose.

Temporal Cohesion

The elements of the module are grouped because they are executed at the same time, such as initialization or cleanup operations.

Logical Cohesion

The elements of the module are related only by their category or type. A module might contain different functions that are logically related but perform different actions.

Coincidential Cohesion

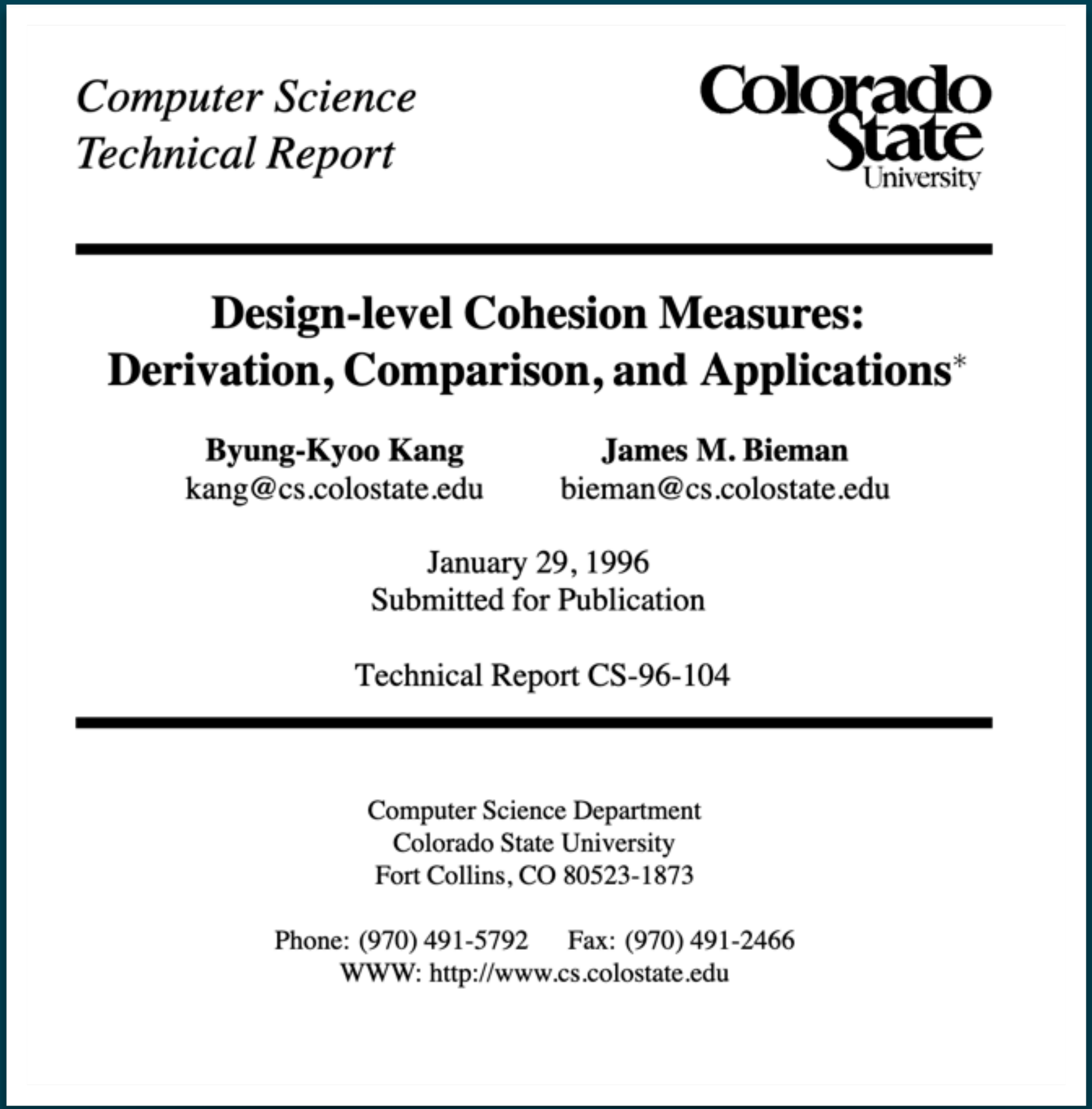
The elements of the module have no meaningful relationship with each other. They are grouped together arbitrarily, often as a result of being developed at the same time.

Wayne P. Stevens, Glenford J. Myers, Larry L. Constantine, **Structured Design**, IBM Systems Journal, 13(2):115-139, 1974.
* was added later

6 Levels of DLC

DLC stands for

Design **L**evel **C**ohesion



Computer Science
Technical Report

Colorado State University

Design-level Cohesion Measures:
Derivation, Comparison, and Applications*

Byung-Kyoo Kang
kang@cs.colostate.edu

James M. Bieman
bieman@cs.colostate.edu

January 29, 1996
Submitted for Publication

Technical Report CS-96-104

Computer Science Department
Colorado State University
Fort Collins, CO 80523-1873

Phone: (970) 491-5792 Fax: (970) 491-2466
WWW: http://www.cs.colostate.edu

6 Levels

of

Design Level Cohesion (DLC)

Functional	There is only one output in a module.
Sequential relation	One output is dependent on the other output.
Communication relation	Two outputs are dependent on a common input. One of the two inputs has data dependence on the input and the other can have a control or a data dependence.
Iterative relation	Two outputs are iteration-control dependent on a common input.
Conditional relation	*Two outputs are condition-control dependent on a common input, or one of two outputs has condition-control dependence on the input and the other has iteration-control dependence on the input.*
Coincidental relation	*Two outputs o1 and o2 of a module have neither dependence relationship with each other, nor dependence on a common input.*

Kang, Bieman: Design-level Cohesion Measures: Derivation, Comparison, and Applications. Colorado State University Technical Report CS-96-104, 1996
https://www.cs.colostate.edu/TechReports/Reports/1996/tr96-104.pdf

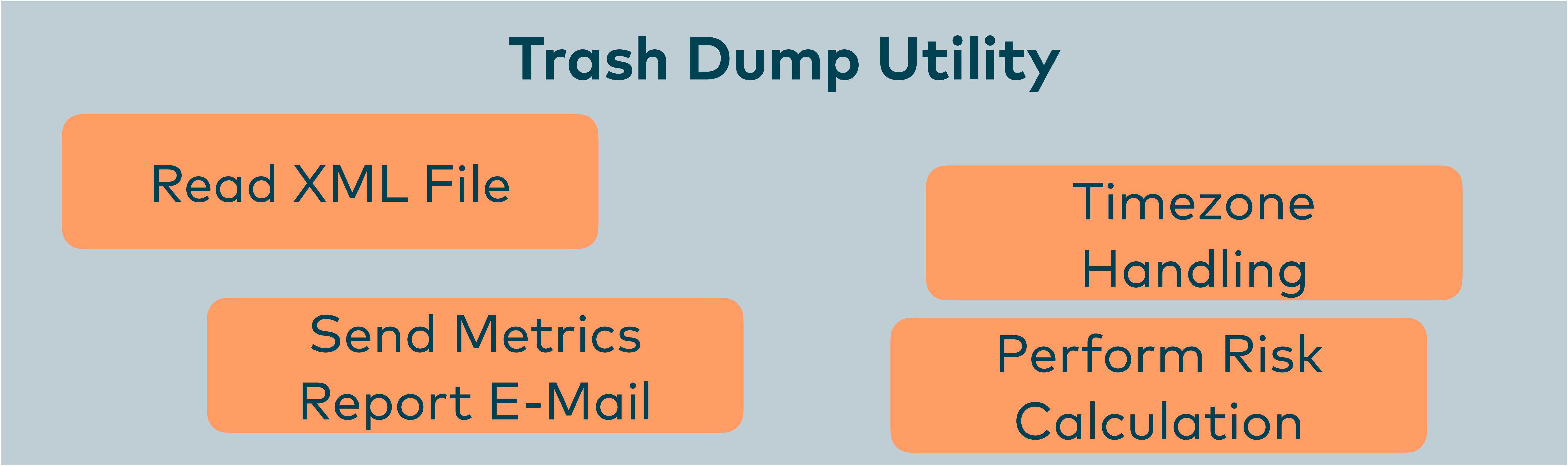
SMC and DLC relate to each other and are very similar

We will continue with SMC for the rest of the talk

Level 1 of 7: Coincidential Cohesion

The elements of the module have **no meaningful relationship** with each other. They are grouped together arbitrarily, often as a result of being developed at the same time.

Example: A utility module that contains a collection of unrelated functions, like formatting text, calculating a sum, and reading a file.



Trash Dump Utility

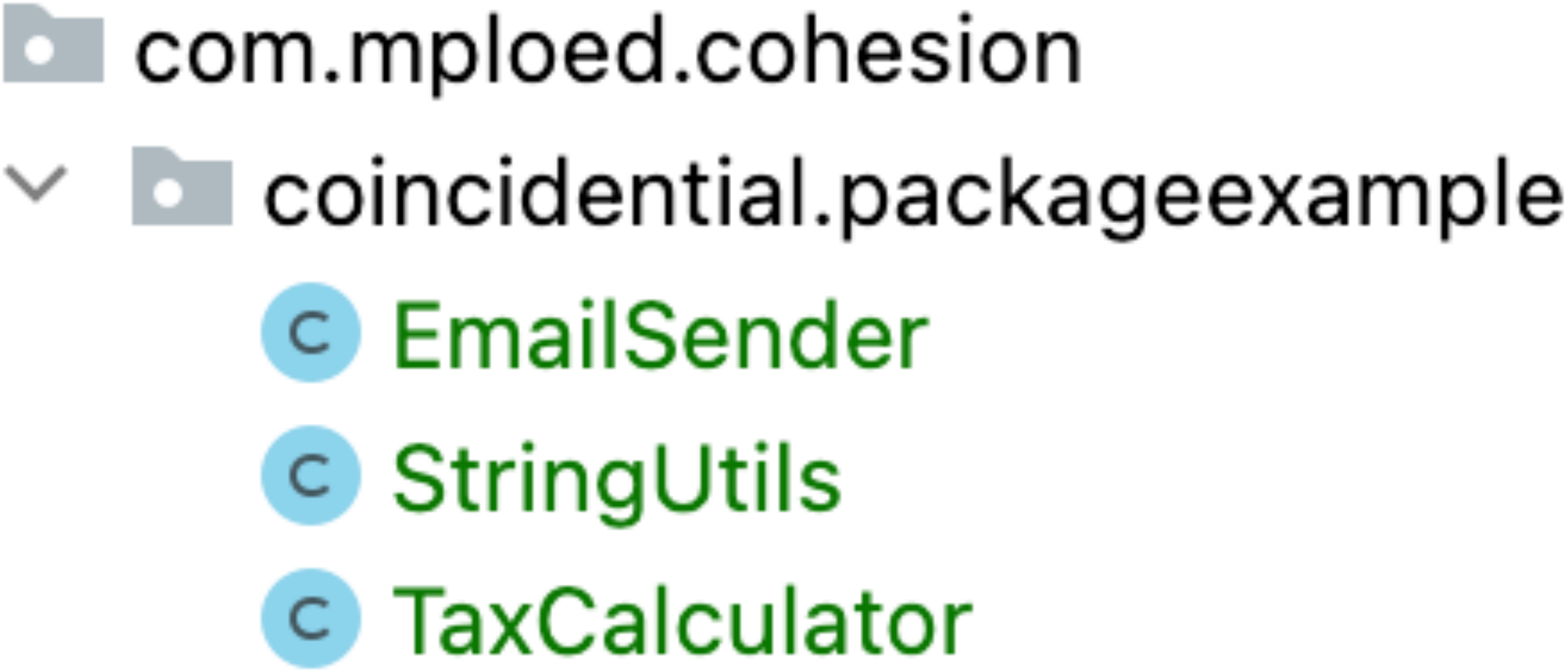
Read XML File

Timezone Handling

Send Metrics Report E-Mail

Perform Risk Calculation

Level 1 of 7: Coincidential Cohesion



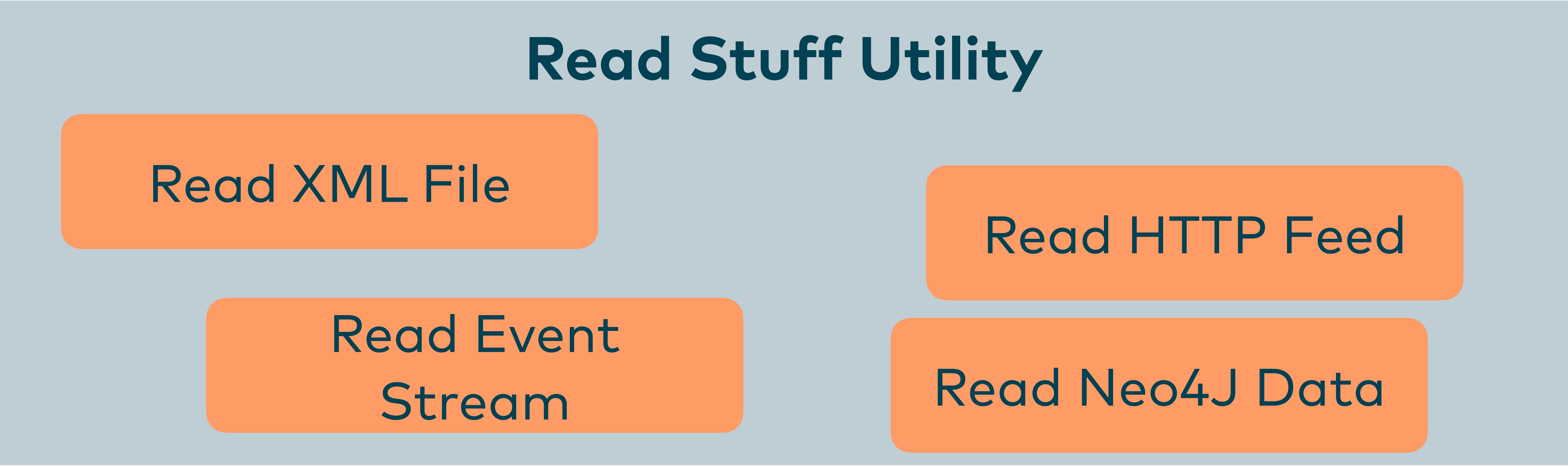
```
com.mploed.cohesion
∨ coincidential.packageexample
    C EmailSender
    C StringUtils
    C TaxCalculator
```

https://github.com/mploed/cohesion-examples

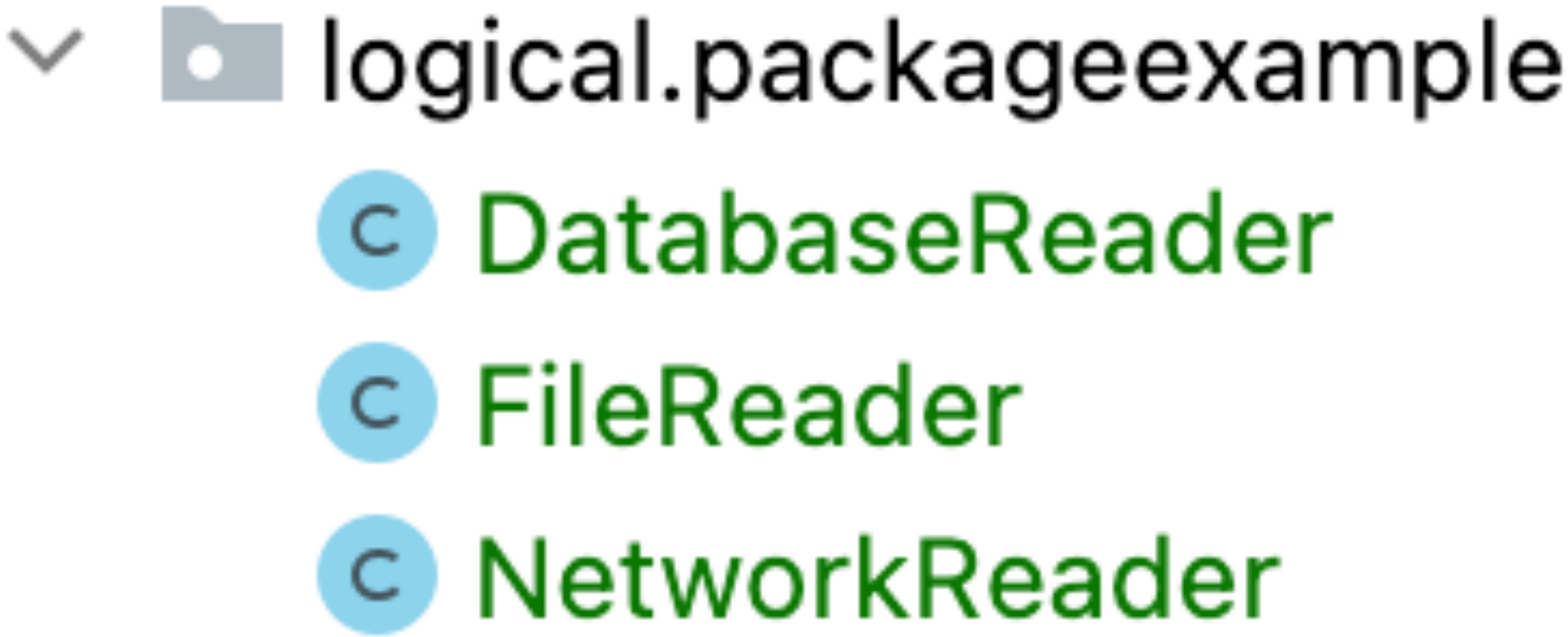
Level 2 of 7: Logical Cohesion

The elements of the module are **related only by their category or type**. A module might contain different functions that are logically related but perform different actions.

Example: A module containing multiple I/O operations such as reading from a file, reading from a database, and reading from a network, reading from a message broker, all grouped because they are "read" operations.



Level 2 of 7: Logical Cohesion



logical.packageexample

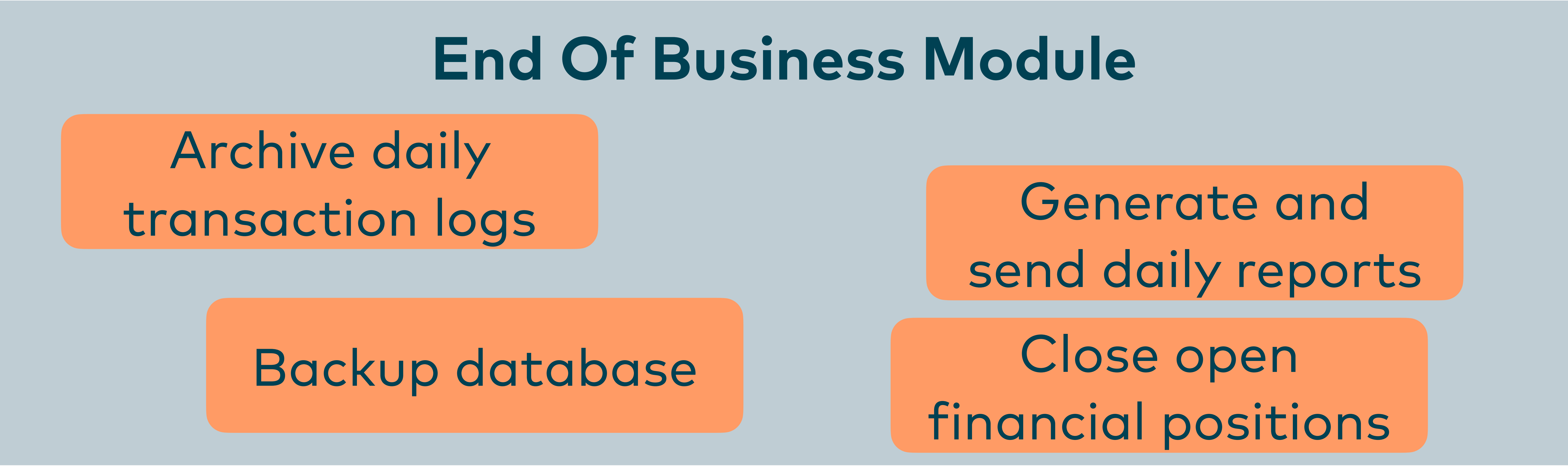
- DatabaseReader
- FileReader
- NetworkReader

https://github.com/mploed/cohesion-examples

Level 3 of 7: Temporal Cohesion

The elements of the module are grouped because they are **executed at the same time but actually they are unrelated** to each other. Characteristics of Logical Cohesion + relationship in time

Example: Operations are performed at the end of the business day, though they serve different purposes (logging, reporting, and data backup).



End Of Business Module

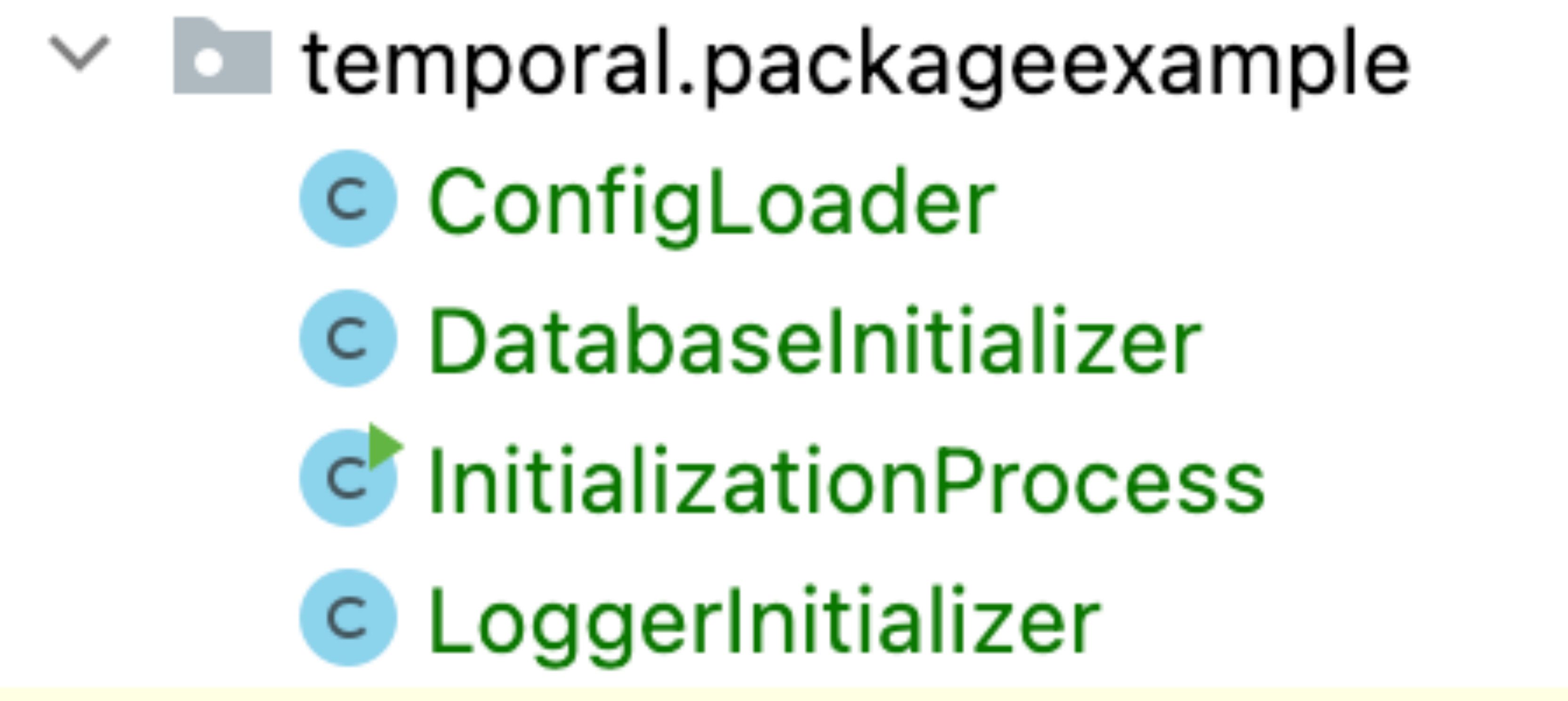
Archive daily transaction logs

Generate and send daily reports

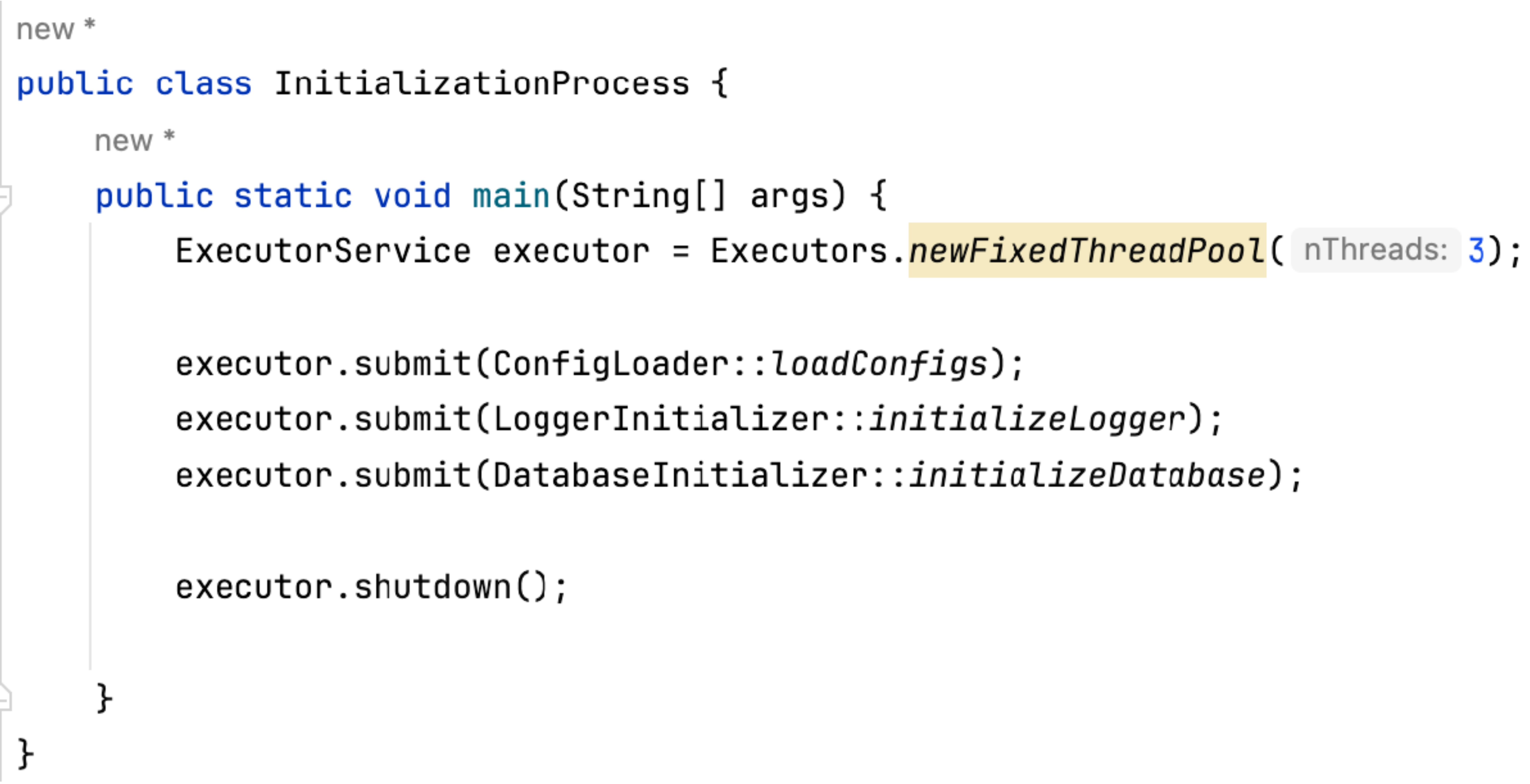
Backup database

Close open financial positions

Level 3 of 7: Temporal Cohesion



- temporal.packageexample
  - ConfigLoader
  - DatabaseInitializer
  - InitializationProcess
  - LoggerInitializer



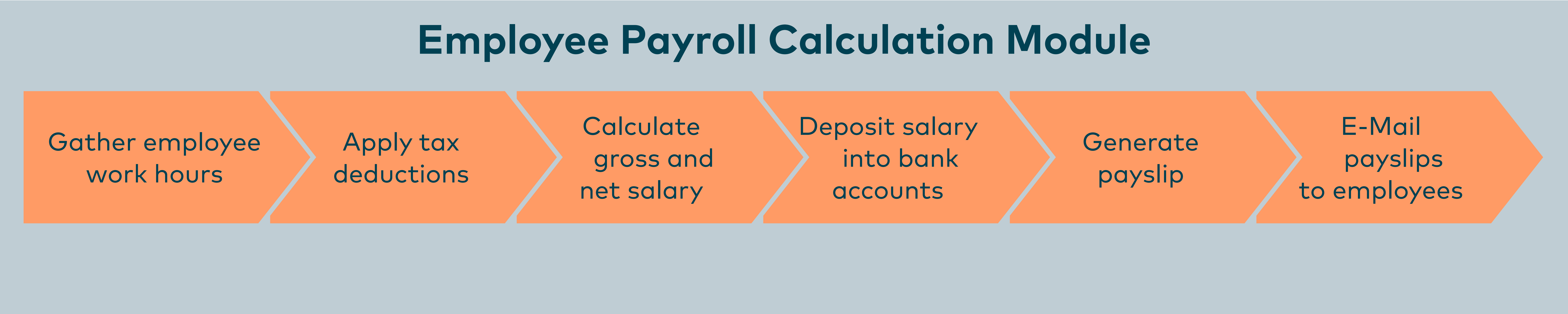
```
new *
public class InitializationProcess {
    new *
    public static void main(String[] args) {
        ExecutorService executor = Executors.newFixedThreadPool( nThreads: 3);

        executor.submit(ConfigLoader::loadConfigs);
        executor.submit(LoggerInitializer::initializeLogger);
        executor.submit(DatabaseInitializer::initializeDatabase);

        executor.shutdown();

    }
}
```

https://github.com/mploed/cohesion-examples

Level 4 of 7: Procedural Cohesion

The elements of the module are grouped because they always follow a **specific sequence of execution**, even if they are not closely related by purpose.

Example: Steps must be performed in a specific order during payroll processing, but they cover different concerns such as financial calculations, tax management, and payment handling.



Employee Payroll Calculation Module

Gather employee work hours

Apply tax deductions

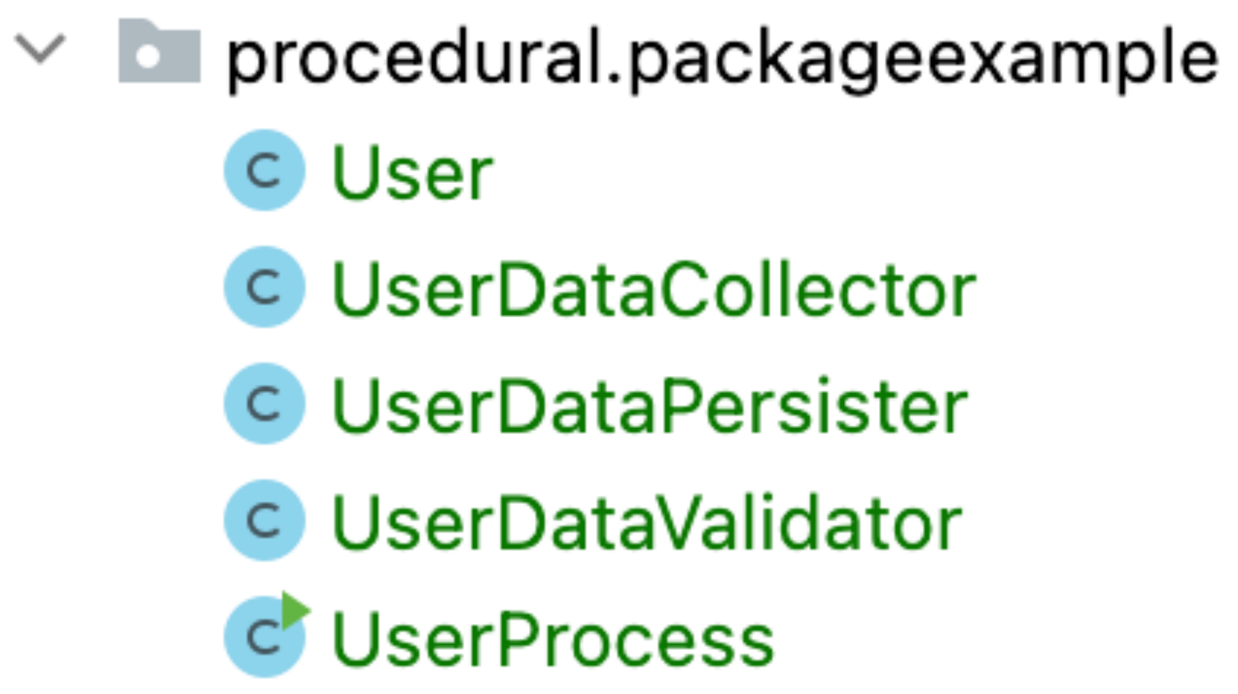
Calculate gross and net salary

Deposit salary into bank accounts

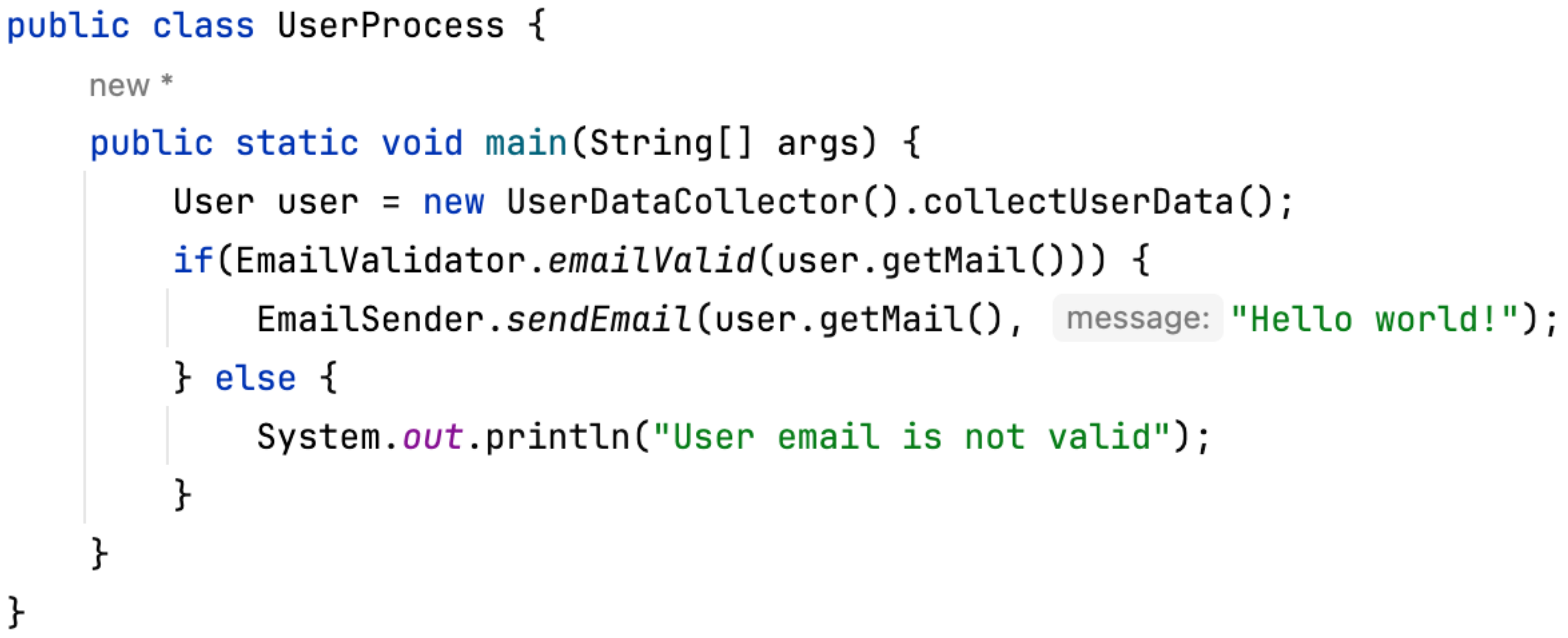
Generate payslip

E-Mail payslips to employees

Level 4 of 7: Procedural Cohesion



- procedural.packageexample
  - User
  - UserDataCollector
  - UserDataPersister
  - UserDataValidator
  - UserProcess



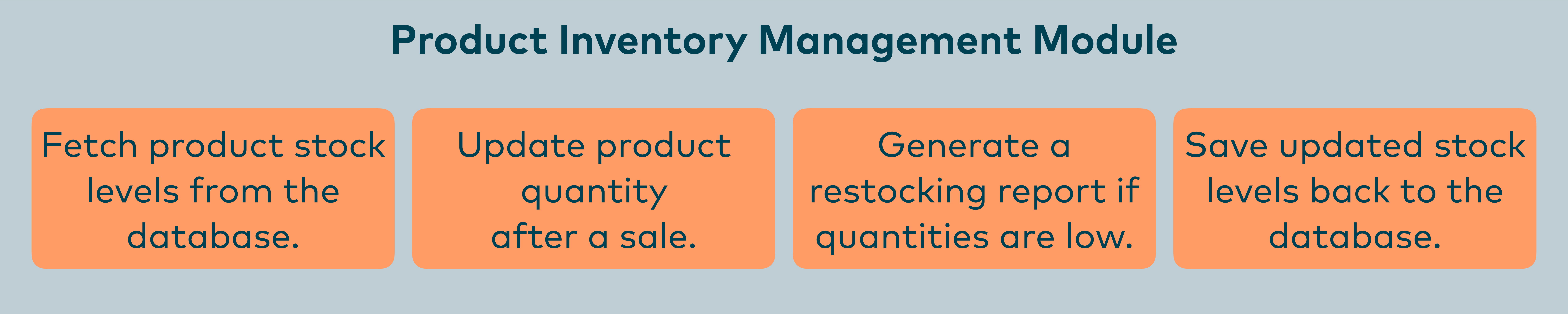
```
public class UserProcess {
    new *
    public static void main(String[] args) {
        User user = new UserDataCollector().collectUserData();
        if(EmailValidator.emailValid(user.getMail())) {
            EmailSender.sendEmail(user.getMail(),  message: "Hello world!");
        } else {
            System.out.println("User email is not valid");
        }
    }
}
```

https://github.com/mploed/cohesion-examples

Level 5 of 7: Communicational Cohesion

The elements of the module are grouped because they **operate on the same data** or contribute to the same data structure. Stronger bond than lower levels because of reference to the same data.

Example: All operations share the same inventory data, performing various tasks like updating, reporting, and saving the stock levels, but working on the same product inventory.



Product Inventory Management Module

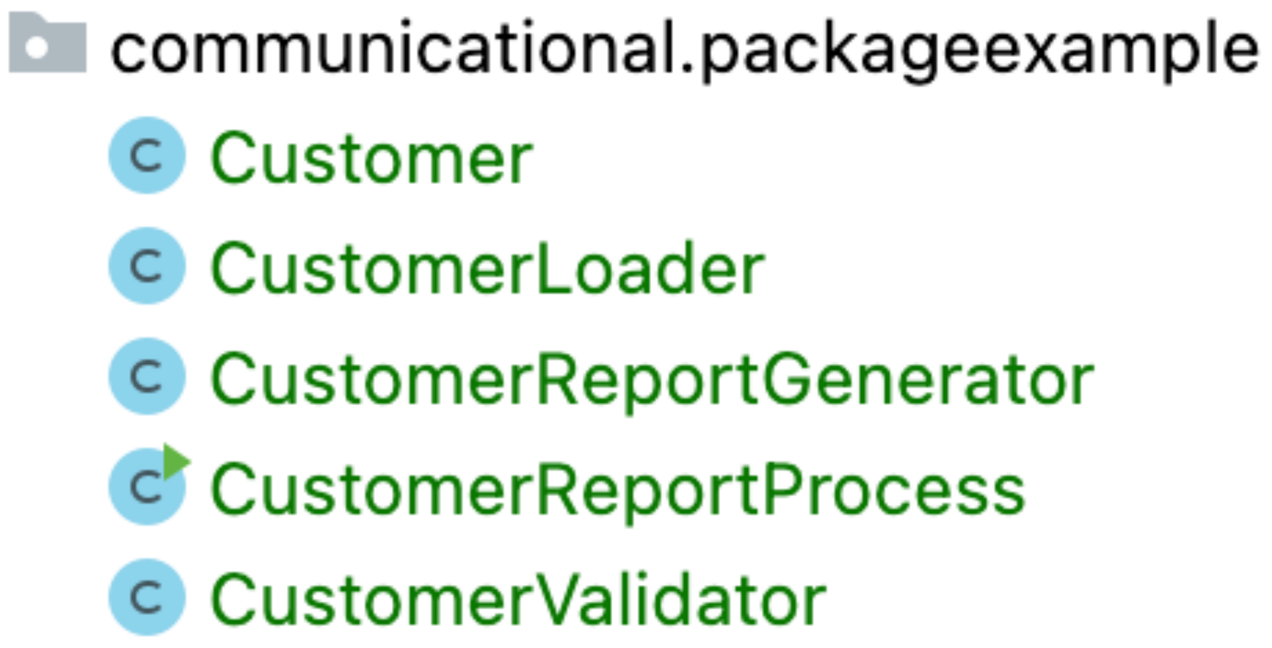
Fetch product stock levels from the database.

Update product quantity after a sale.

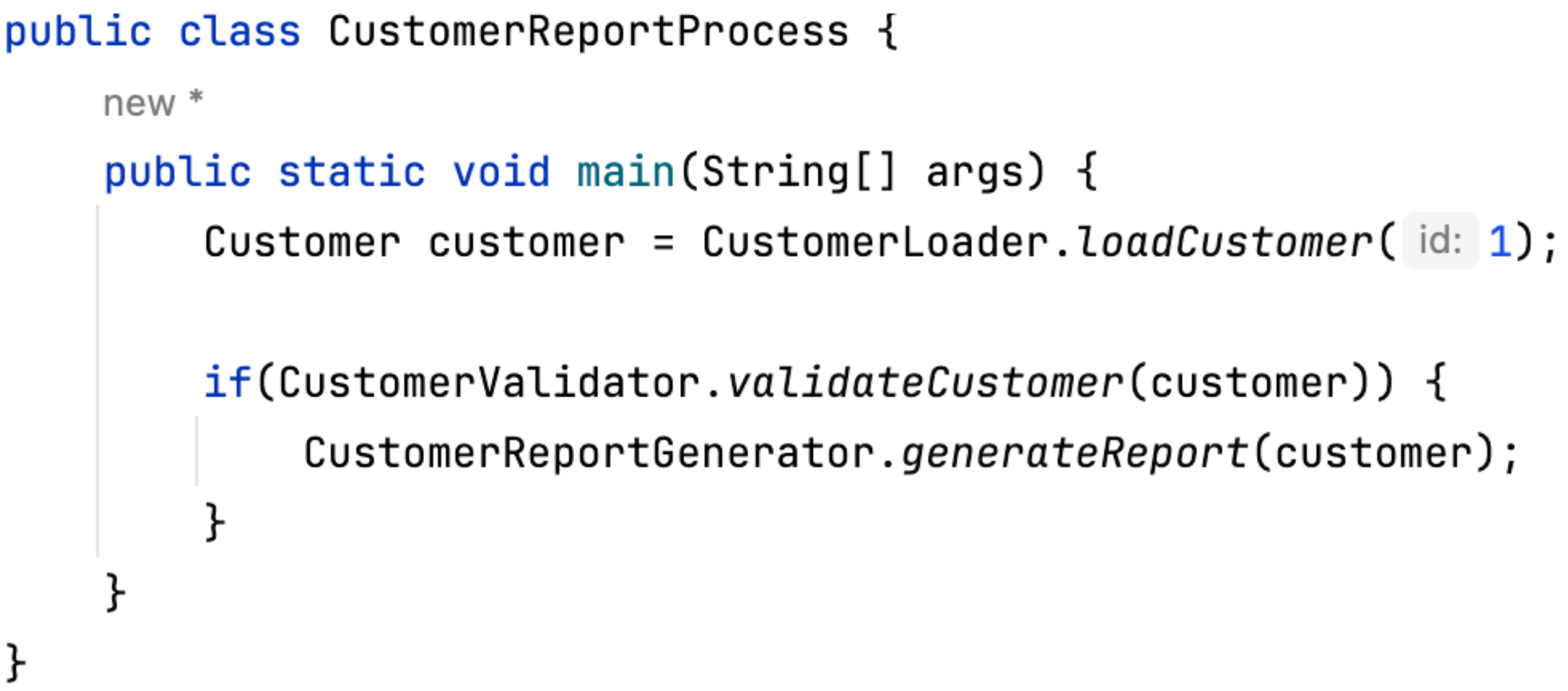
Generate a restocking report if quantities are low.

Save updated stock levels back to the database.

Level 5 of 7: Communicational Cohesion



- communicational.packageexample
  - Customer
  - CustomerLoader
  - CustomerReportGenerator
  - CustomerReportProcess
  - CustomerValidator



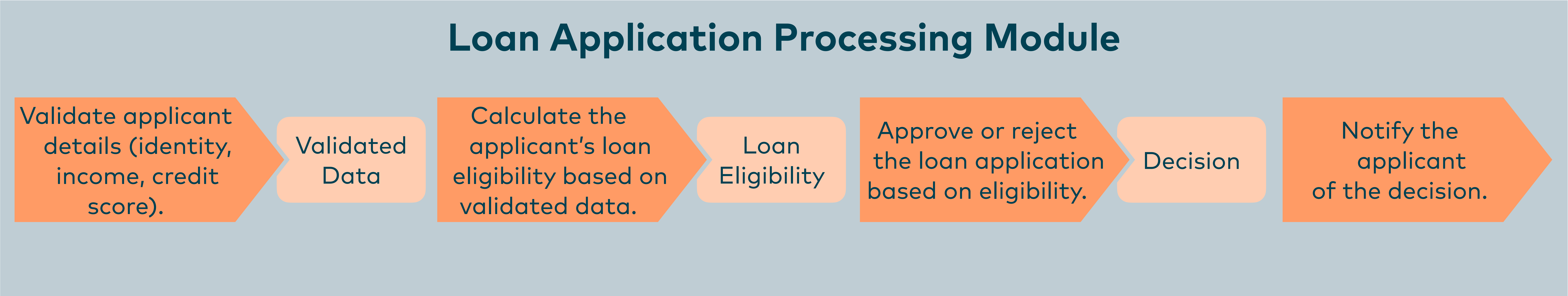
```
public class CustomerReportProcess {
    new *
    public static void main(String[] args) {
        Customer customer = CustomerLoader.loadCustomer( id: 1);

        if(CustomerValidator.validateCustomer(customer)) {
            CustomerReportGenerator.generateReport(customer);
        }
    }
}
```

https://github.com/mploed/cohesion-examples

Level 6 of 7: Sequential Cohesion

The elements of the module are grouped because the **output from one element serves as input to another**. This type of cohesion exists when functions are interdependent in a linear chain.

Example: The loan eligibility calculation requires validated applicant details, and the approval decision is based on the eligibility result. The final step notifies the applicant of the decision, creating a clear sequential flow.



Loan Application Processing Module

Validate applicant details (identity, income, credit score).

Validated Data

Calculate the applicant's loan eligibility based on validated data.

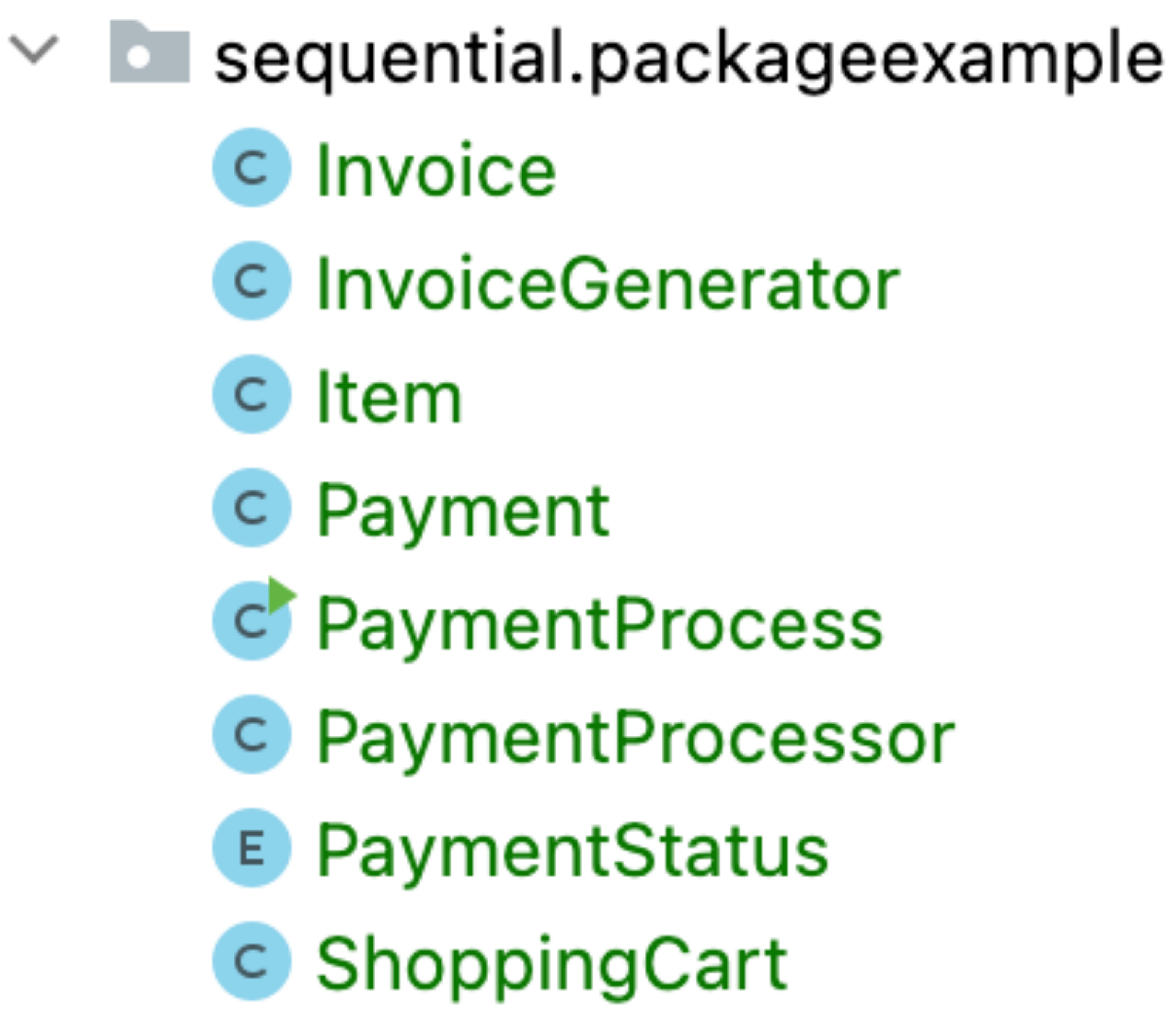
Loan Eligibility

Approve or reject the loan application based on eligibility.

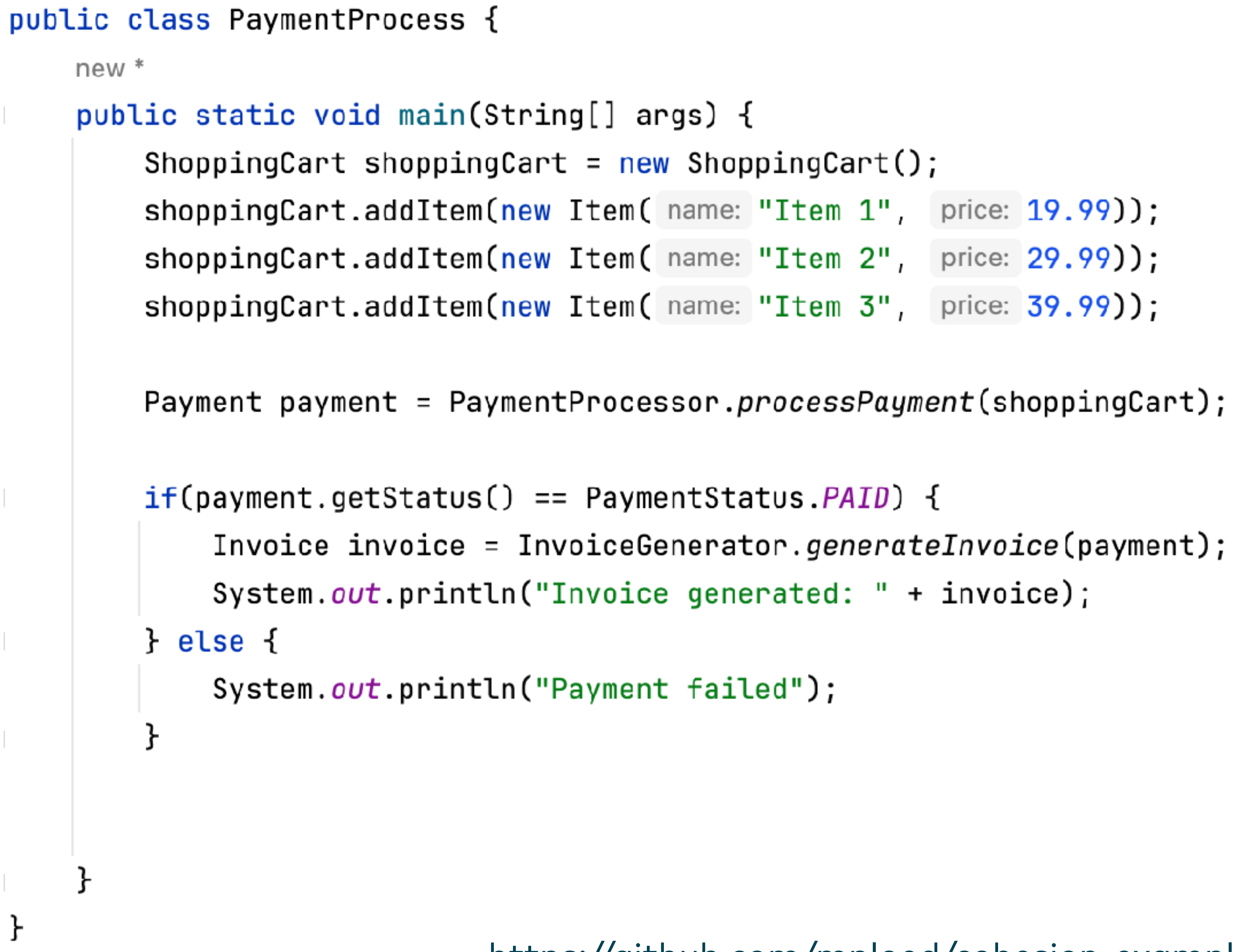
Decision

Notify the applicant of the decision.

Level 6 of 7: Sequential Cohesion



- sequential.packageexample
  - C Invoice
  - C InvoiceGenerator
  - C Item
  - C Payment
  - C PaymentProcess
  - C PaymentProcessor
  - E PaymentStatus
  - C ShoppingCart



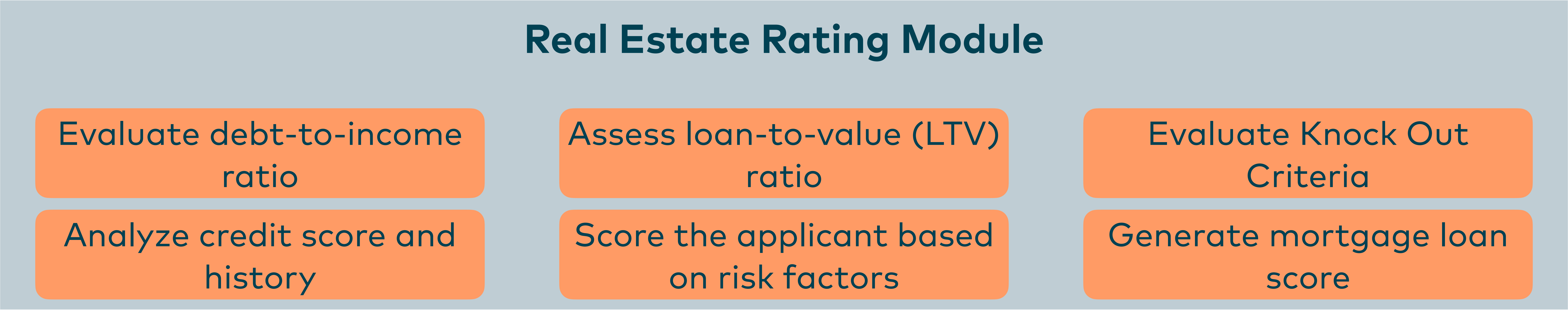
```java
public class PaymentProcess {
    new *
    public static void main(String[] args) {
        ShoppingCart shoppingCart = new ShoppingCart();
        shoppingCart.addItem(new Item( name: "Item 1",  price: 19.99));
        shoppingCart.addItem(new Item( name: "Item 2",  price: 29.99));
        shoppingCart.addItem(new Item( name: "Item 3",  price: 39.99));

        Payment payment = PaymentProcessor.processPayment(shoppingCart);

        if(payment.getStatus() == PaymentStatus.PAID) {
            Invoice invoice = InvoiceGenerator.generateInvoice(payment);
            System.out.println("Invoice generated: " + invoice);
        } else {
            System.out.println("Payment failed");
        }


    }
}
```

https://github.com/mploed/cohesion-examples

Level 7 of 7: Functional Cohesion

The elements of the module are grouped because they all **contribute to performing a single, well-defined task**. Every part of the module is essential to its overall functionality.

Example: All operations are centered on the single task of scoring a mortgage loan applicant. Every operation contributes directly to completing the goal of assessing the applicant's eligibility and risk for a mortgage loan.



Real Estate Rating Module

Evaluate debt-to-income ratio

Assess loan-to-value (LTV) ratio

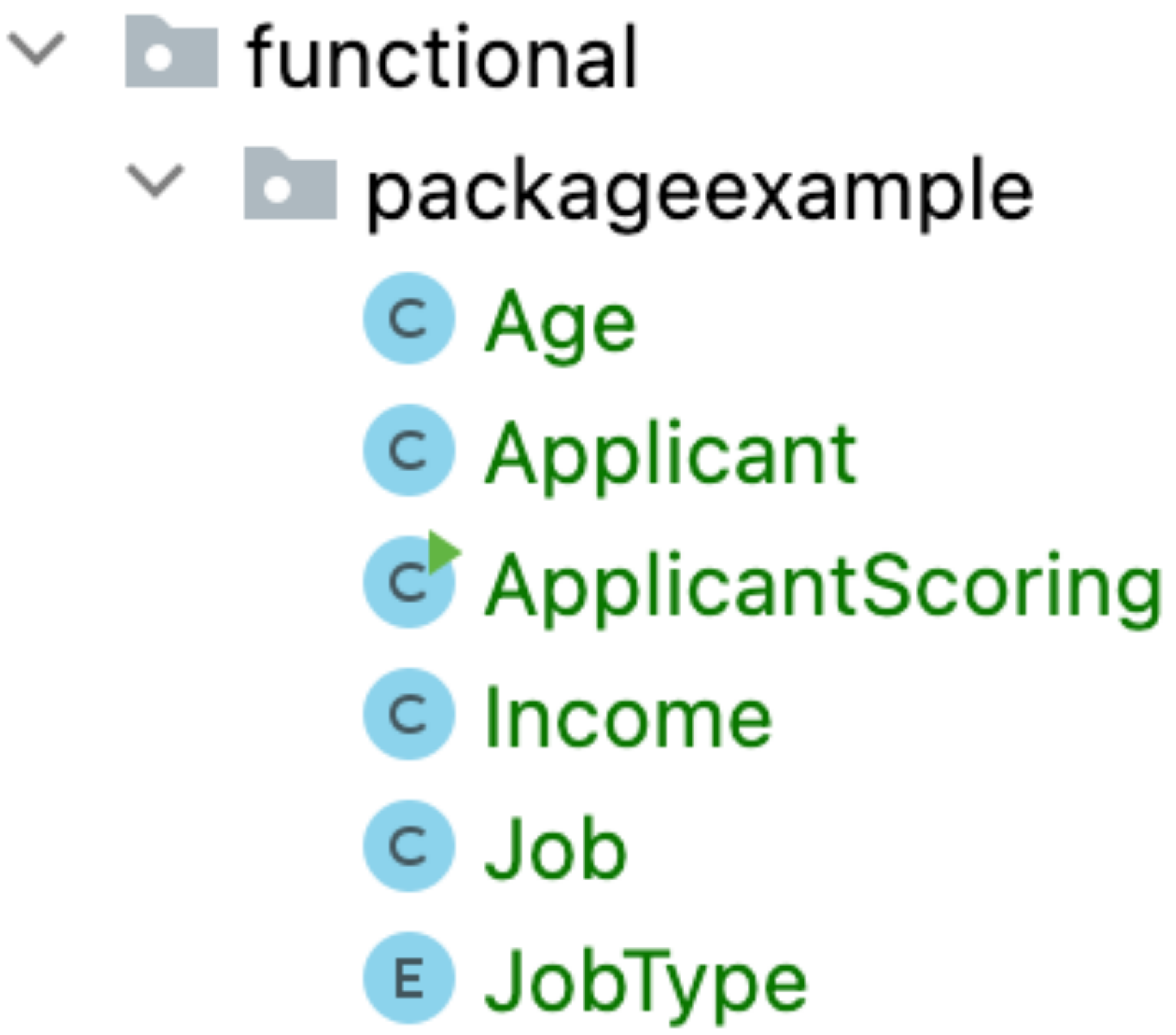
Evaluate Knock Out Criteria

Analyze credit score and history

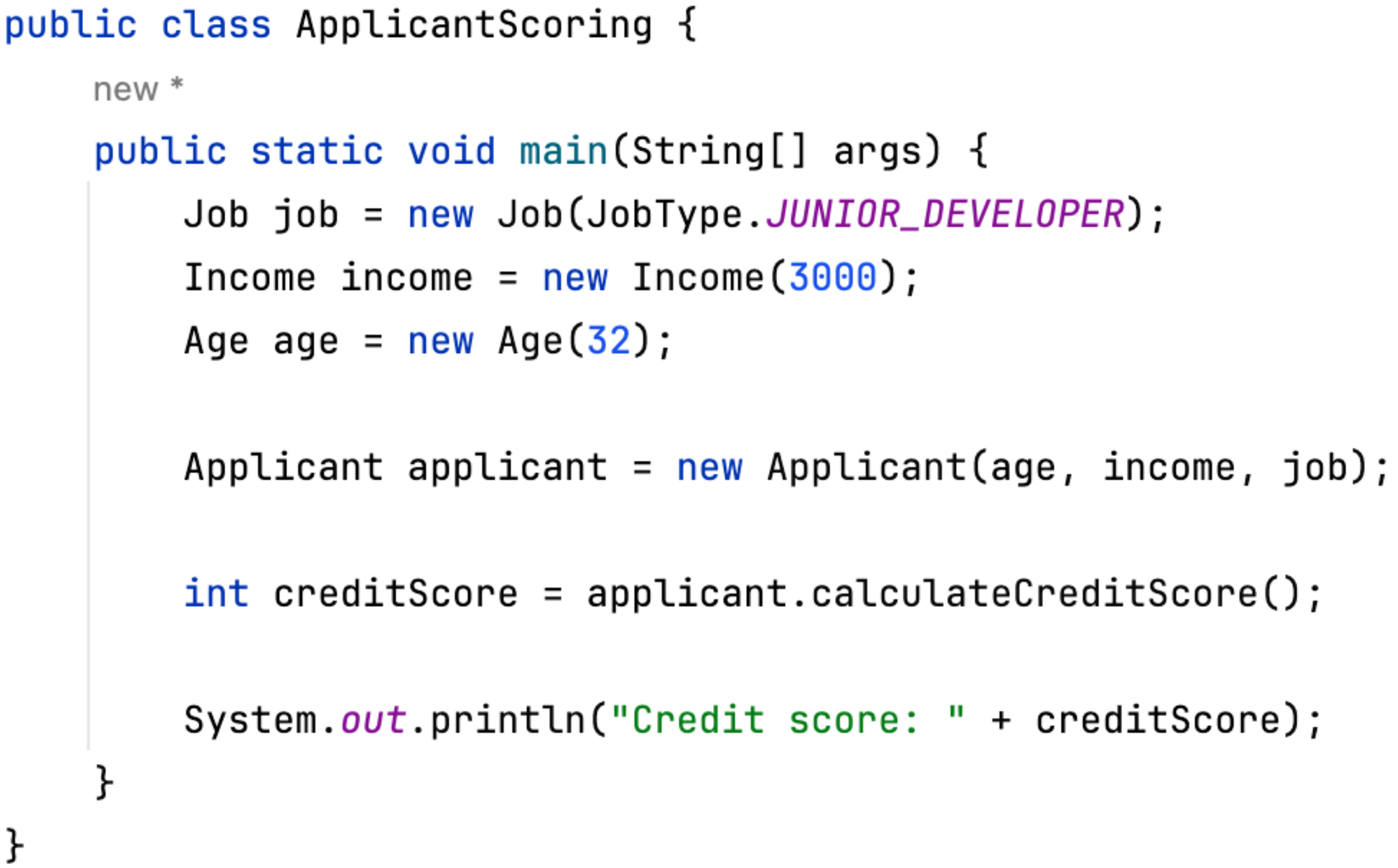
Score the applicant based on risk factors

Generate mortgage loan score

Level 7 of 7: Functional Cohesion



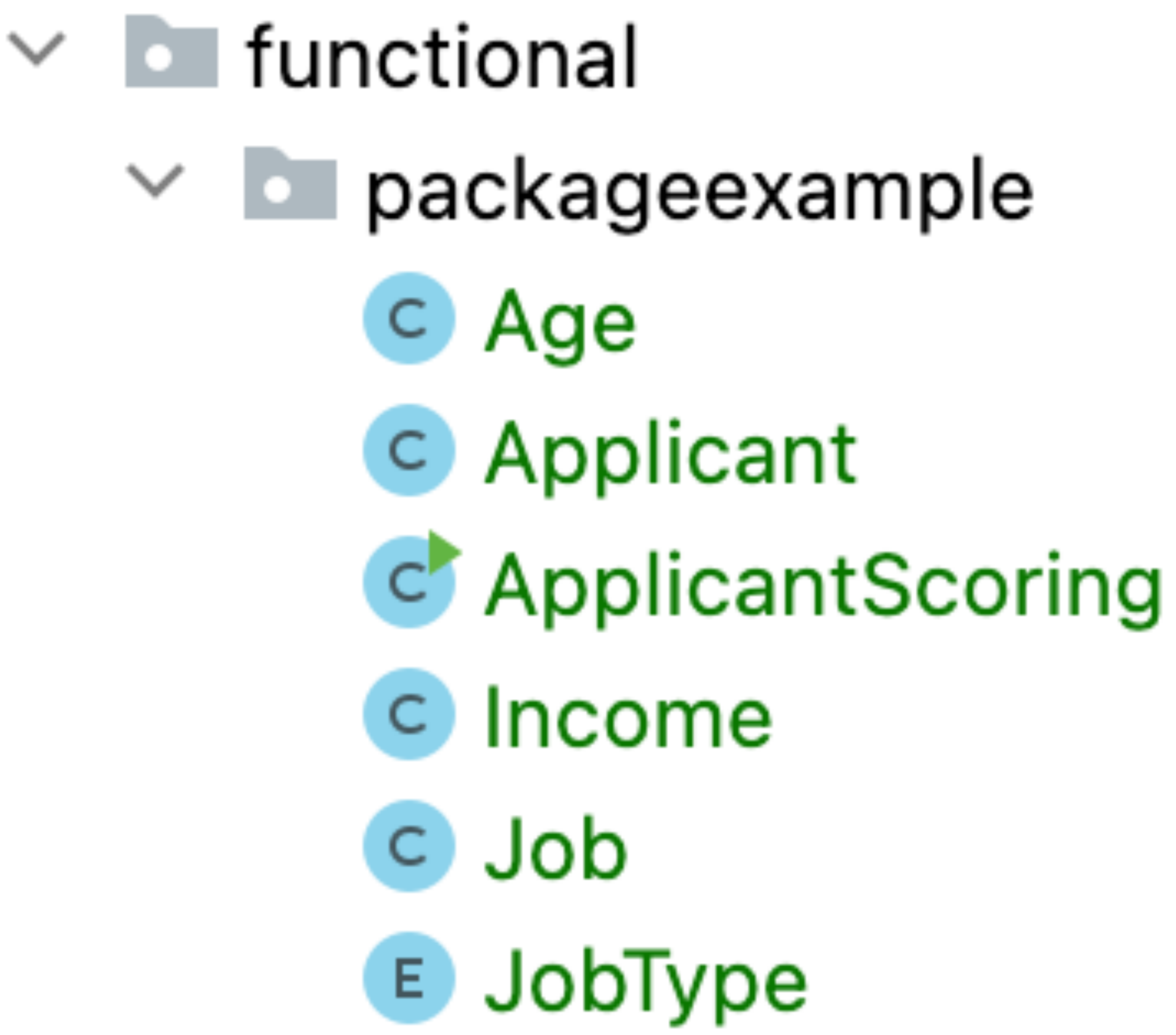
```
functional
  packageexample
    C Age
    C Applicant
    C ApplicantScoring
    C Income
    C Job
    E JobType
```



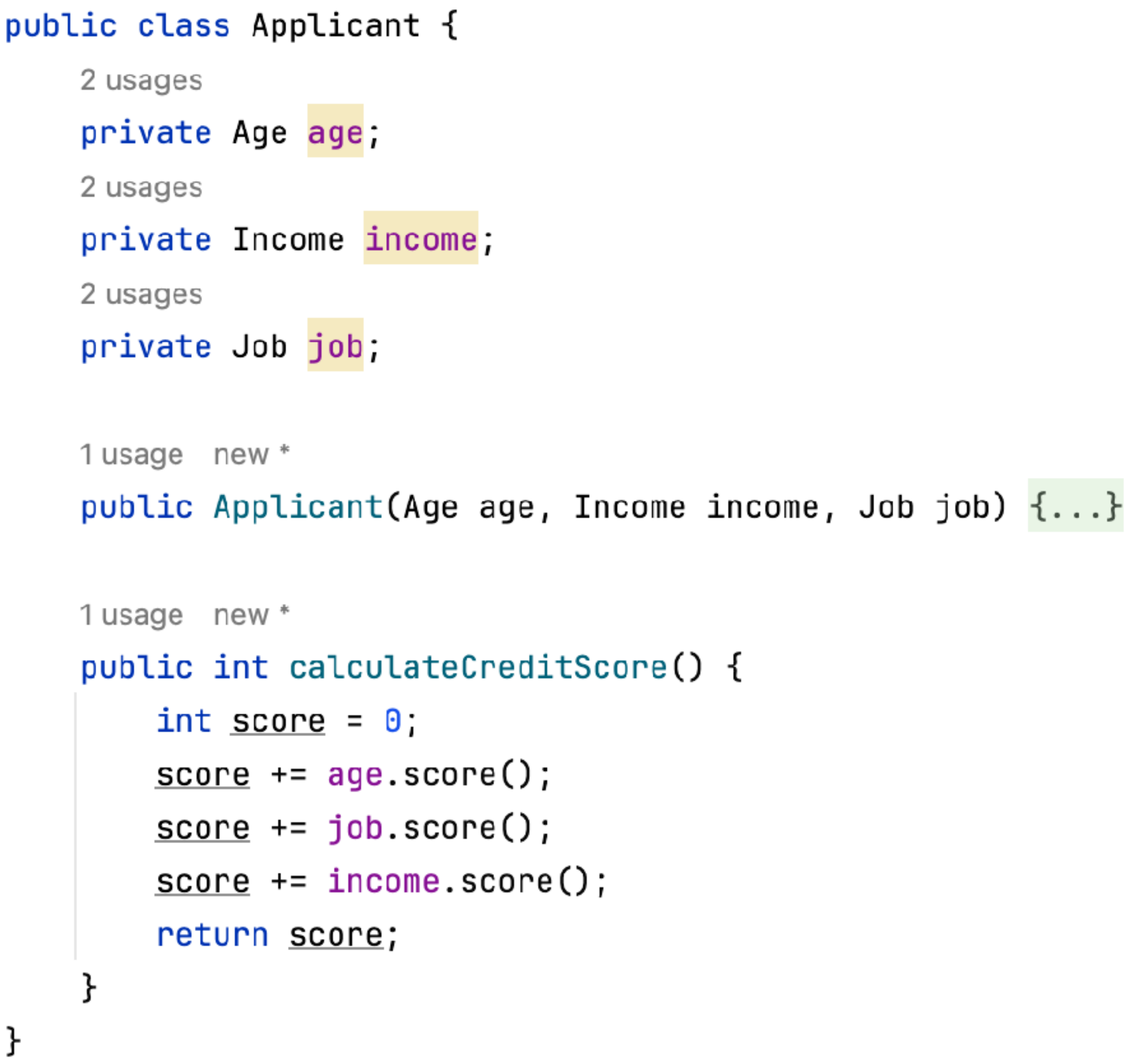
```java
public class ApplicantScoring {
    new *
    public static void main(String[] args) {
        Job job = new Job(JobType.JUNIOR_DEVELOPER);
        Income income = new Income(3000);
        Age age = new Age(32);

        Applicant applicant = new Applicant(age, income, job);

        int creditScore = applicant.calculateCreditScore();

        System.out.println("Credit score: " + creditScore);
    }
}
```

https://github.com/mploed/cohesion-examples

Level 7 of 7: Functional Cohesion continued



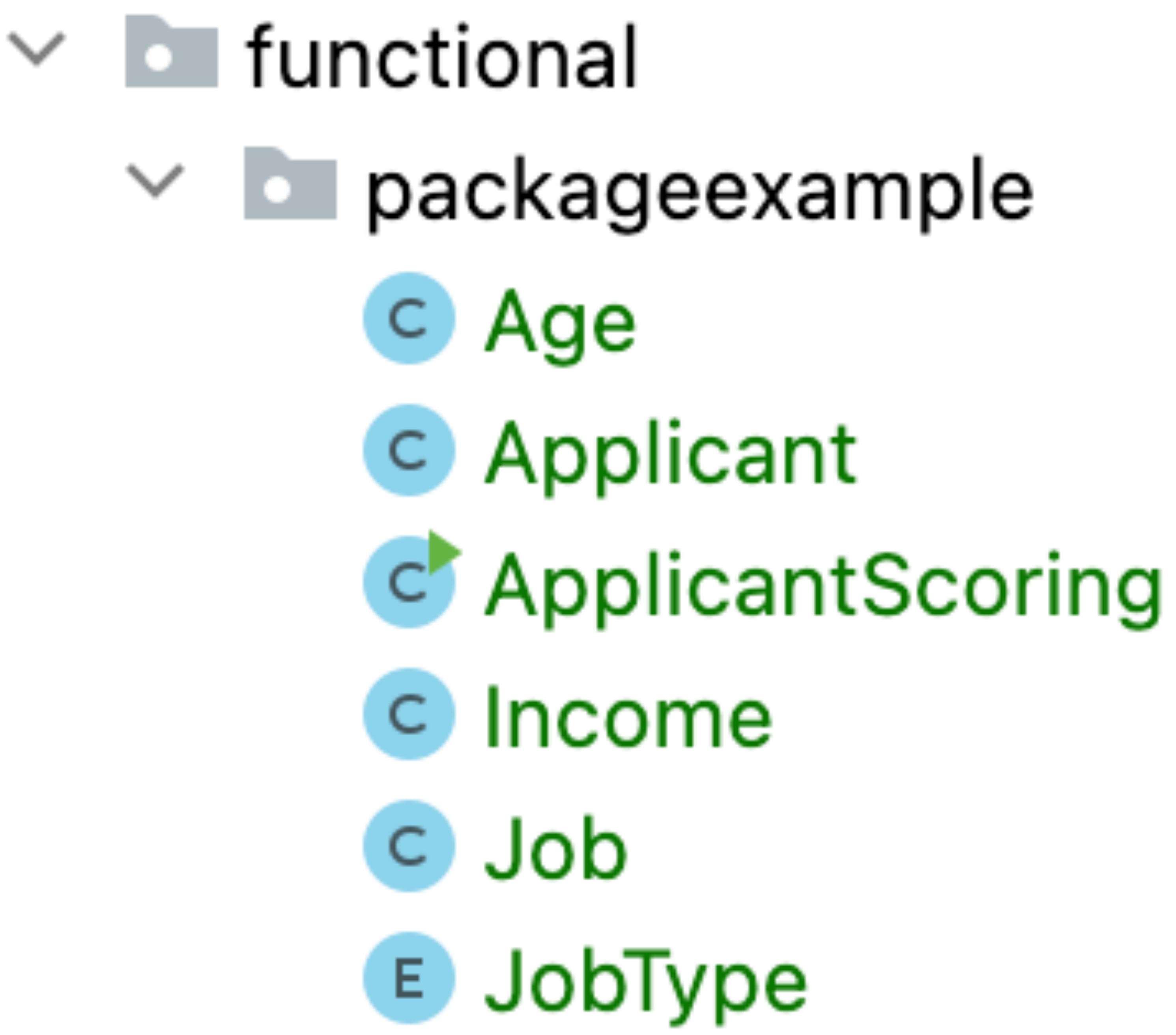
```
functional
  packageexample
    Age
    Applicant
    ApplicantScoring
    Income
    Job
    JobType
```



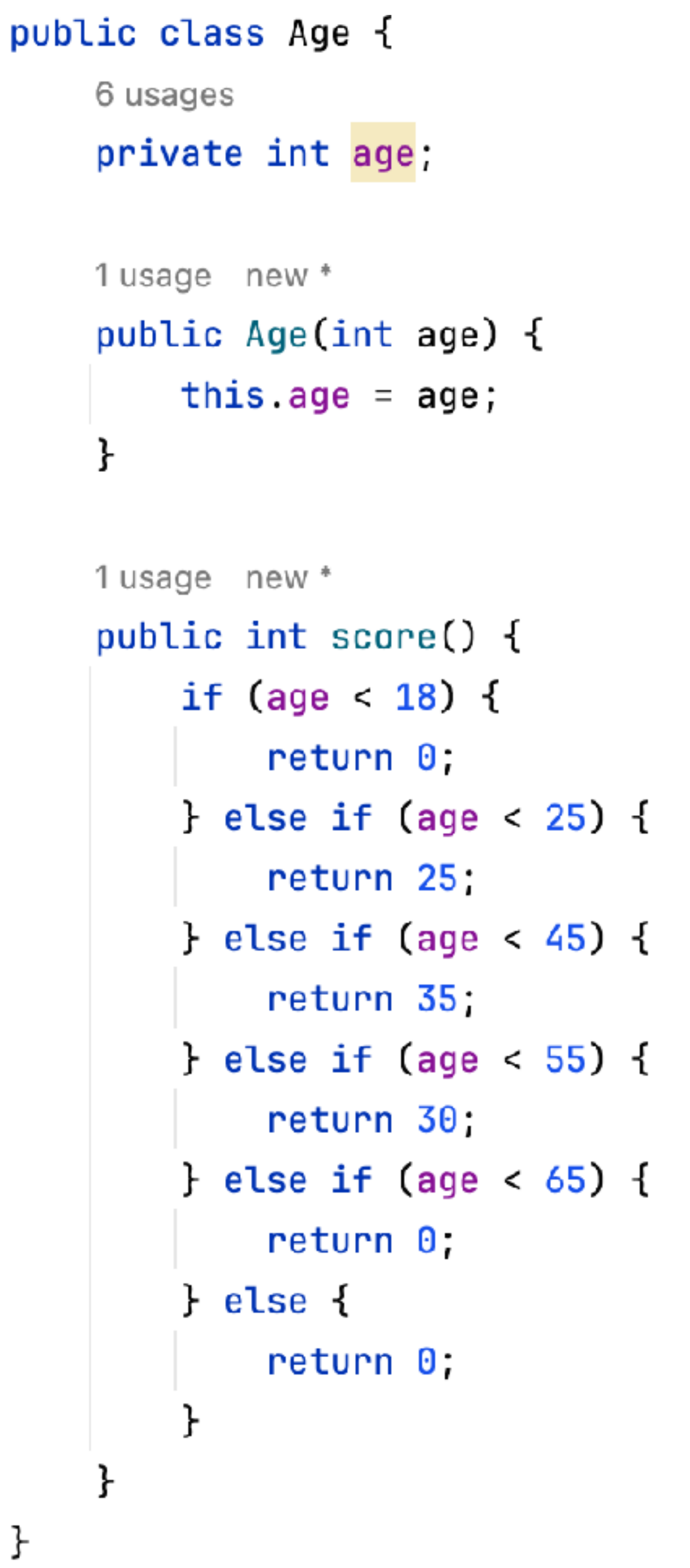
```
public class Applicant {
    2 usages
    private Age age;
    2 usages
    private Income income;
    2 usages
    private Job job;

    1 usage   new *
    public Applicant(Age age, Income income, Job job) {...}

    1 usage   new *
    public int calculateCreditScore() {
        int score = 0;
        score += age.score();
        score += job.score();
        score += income.score();
        return score;
    }
}
```

https://github.com/mploed/cohesion-examples

Level 7 of 7: Functional Cohesion continued



- functional
  - packageexample
    - Age
    - Applicant
    - ApplicantScoring
    - Income
    - Job
    - JobType



```java
public class Age {
    6 usages
    private int age;

    1 usage   new *
    public Age(int age) {
        this.age = age;
    }

    1 usage   new *
    public int score() {
        if (age < 18) {
            return 0;
        } else if (age < 25) {
            return 25;
        } else if (age < 45) {
            return 35;
        } else if (age < 55) {
            return 30;
        } else if (age < 65) {
            return 0;
        } else {
            return 0;
        }
    }
}
```

https://github.com/mploed/cohesion-examples

Metrics

for

Cohesion

Metrics for measuring cohesion

- Lack of Cohesion in Methods (LCOM 1-4)
- Tight Class Cohesion (TCC)
- Information Flow-based Cohesion (ICH)
- Loose Class Cohesion (LCC)
- Class Cohesion (CC)
- Cohesion Among Methods of Class (CAMC)
- Degree of Cohesion (DC)

Two examples: LCOM and TCC

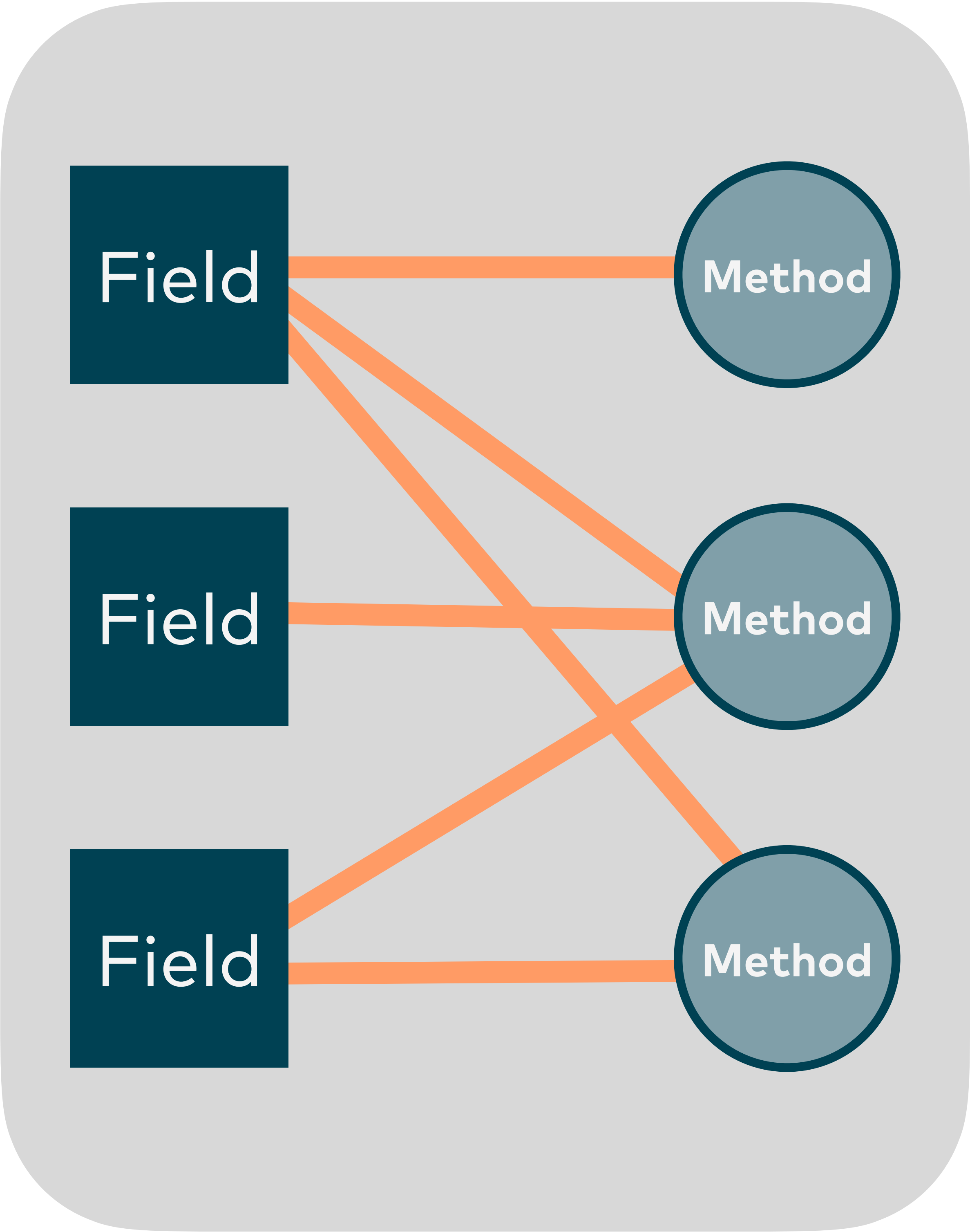
Chidamber & Kemerer LCOM Family

LCOM measures how disjointed the methods of a class are. A high LCOM value indicates that methods are not working on the same data, suggesting low cohesion.

Variants: There are different versions of LCOM, such as LCOM1, LCOM2, LCOM3, etc., with slight variations in how they are calculated.

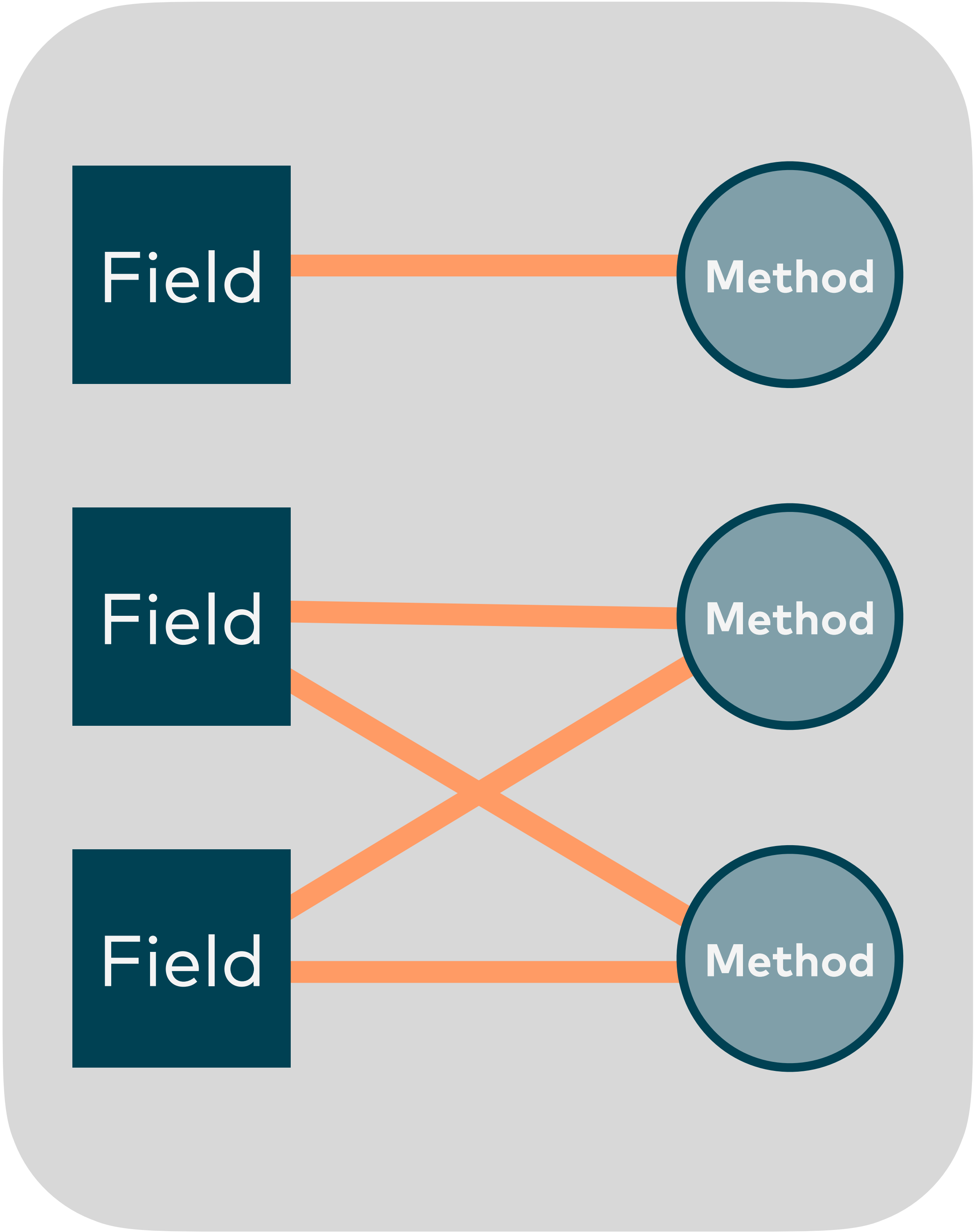
Formula (LCOM1): Count pairs of methods that do not share instance variables and subtract from the pairs that do. A higher result indicates lower cohesion.

LCOM Family



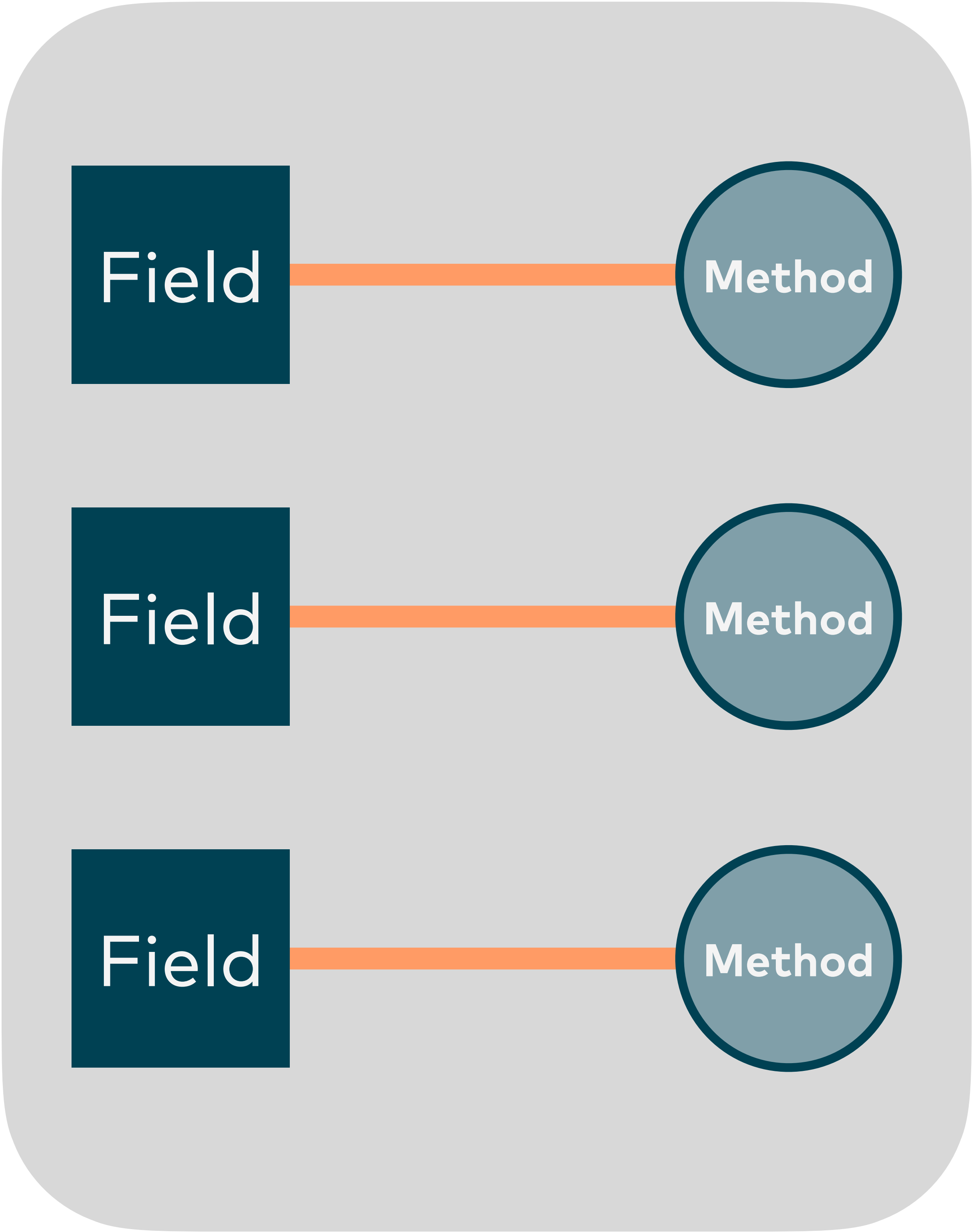
Low LCOM score (0)

High Cohesion



Medium LCOM score (1)

Moderate / Mixed Cohesion



High LCOM score (3)

Low Cohesion

Inspired by Richards, Ford. Visualization in Fundamentals of Software Architecture p. 43, O'Reilly 2020

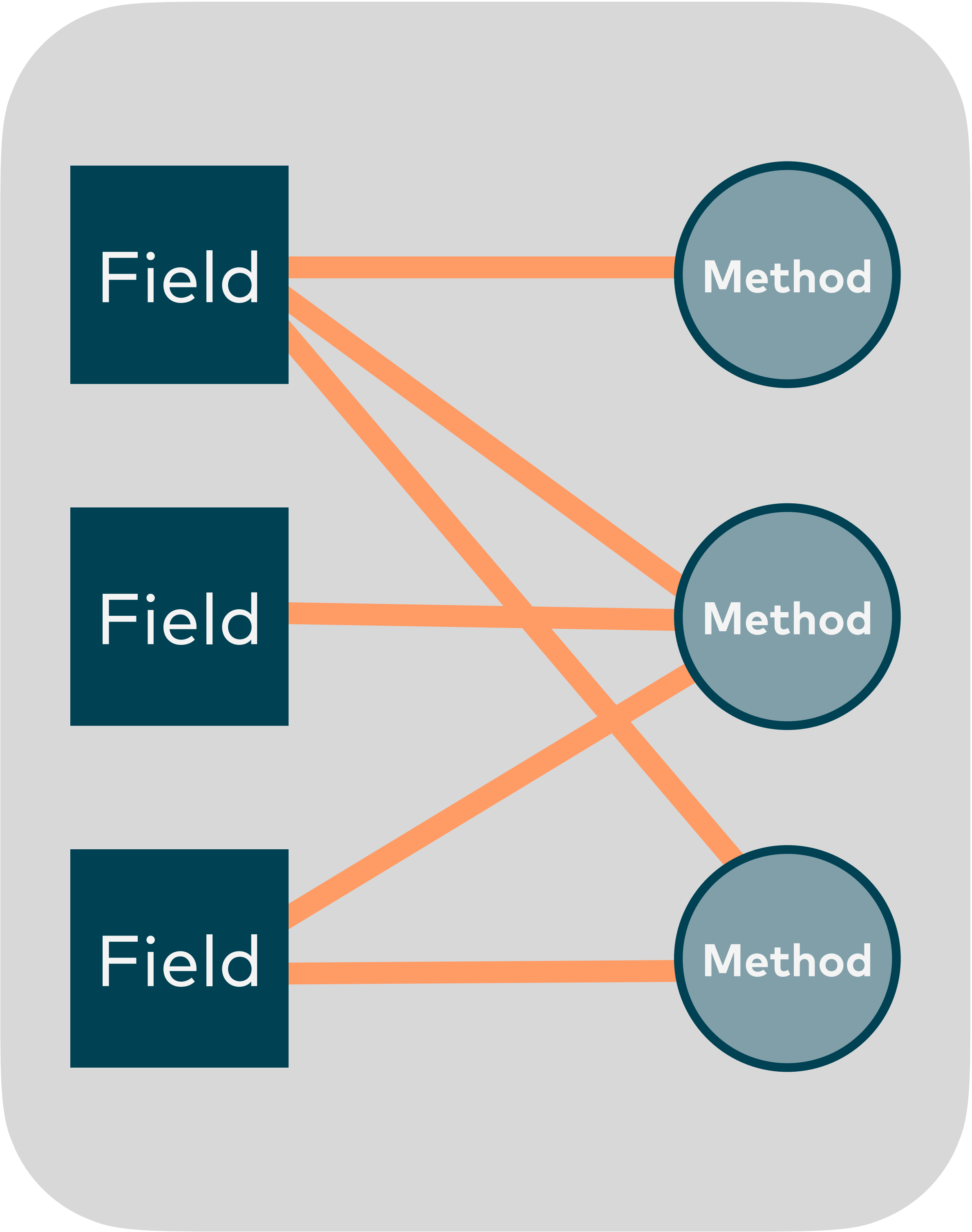
Tight Class Cohesion (TCC)

TCC measures the ratio of pairs of methods that access at least one common attribute to the total number of method pairs. A higher TCC indicates higher cohesion. This metric provides a fine-grained measure of how related the methods of a class are through shared data.

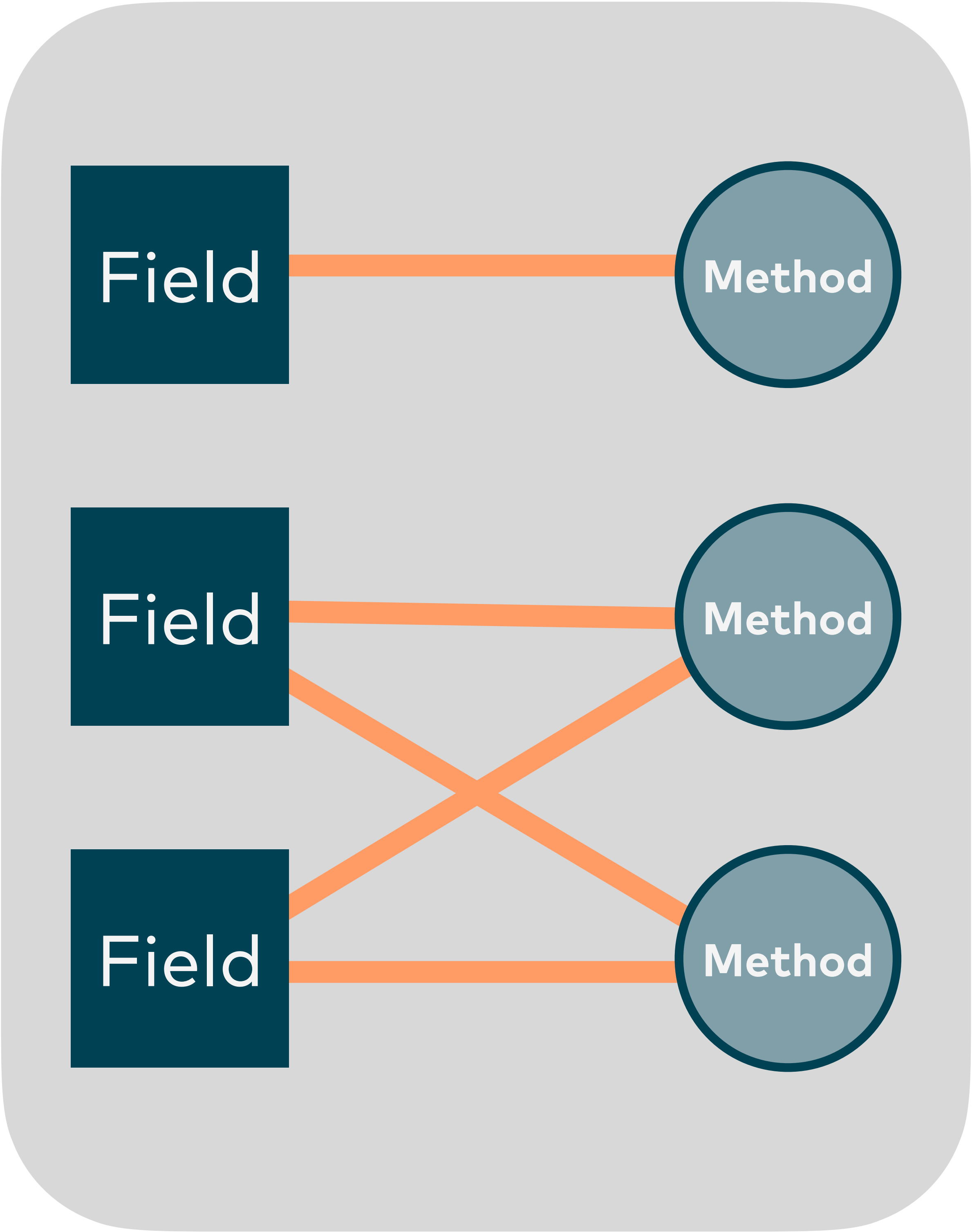
Formula: TCC = (number of method pairs that share attributes) / (total number of method pairs).

Range: TCC values range from 0 (low cohesion) to 1 (high cohesion).

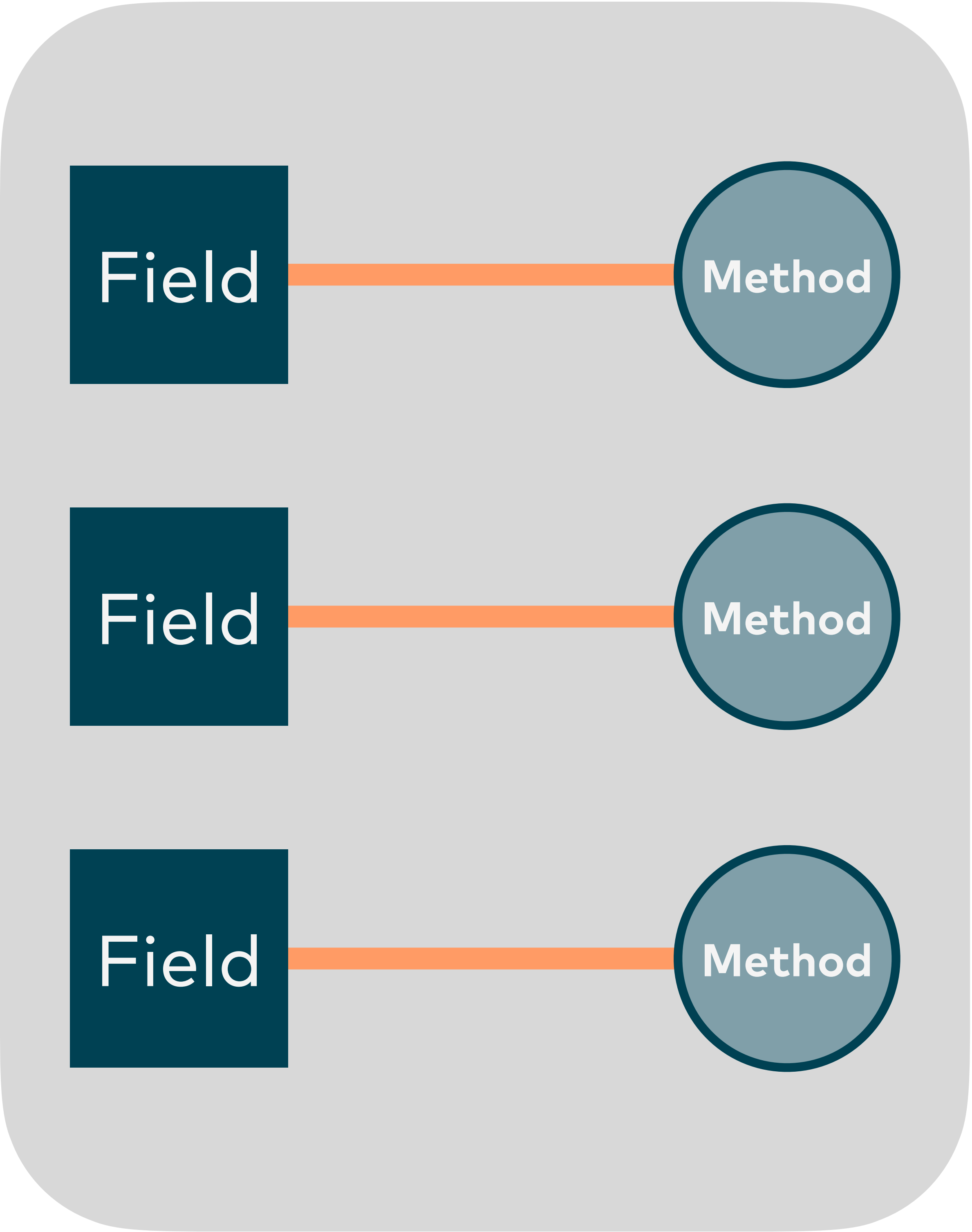
TCC



1 (3 / 3)



0,67 (2 / 3)



0 (0 / 3)

Inspired by Richards, Ford. Visualization in Fundamentals of Software Architecture p. 43, O'Reilly 2020

Don't blindly believe in metrics!

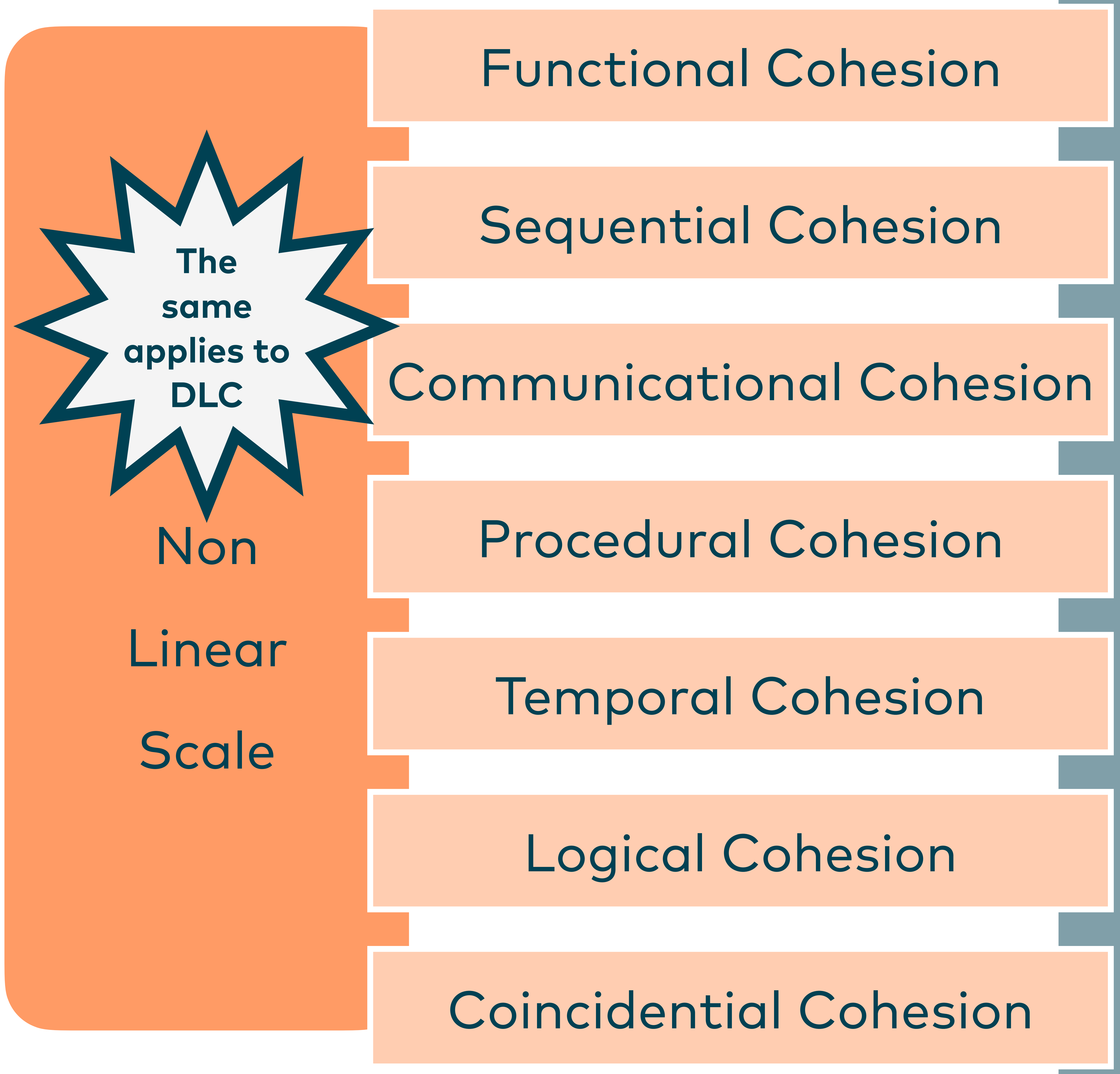
Context matters

The evolution of their results is more interesting than a temporal snapshot

The golden rule of metrics

Never show metrics to non-IT folks

EVER!



Functional Cohesion

Sequential Cohesion

Communicational Cohesion

Procedural Cohesion

Temporal Cohesion

Logical Cohesion

Coincidential Cohesion

The same applies to DLC

Non

Linear

Scale

Functional Cohesion is stronger than all of the others

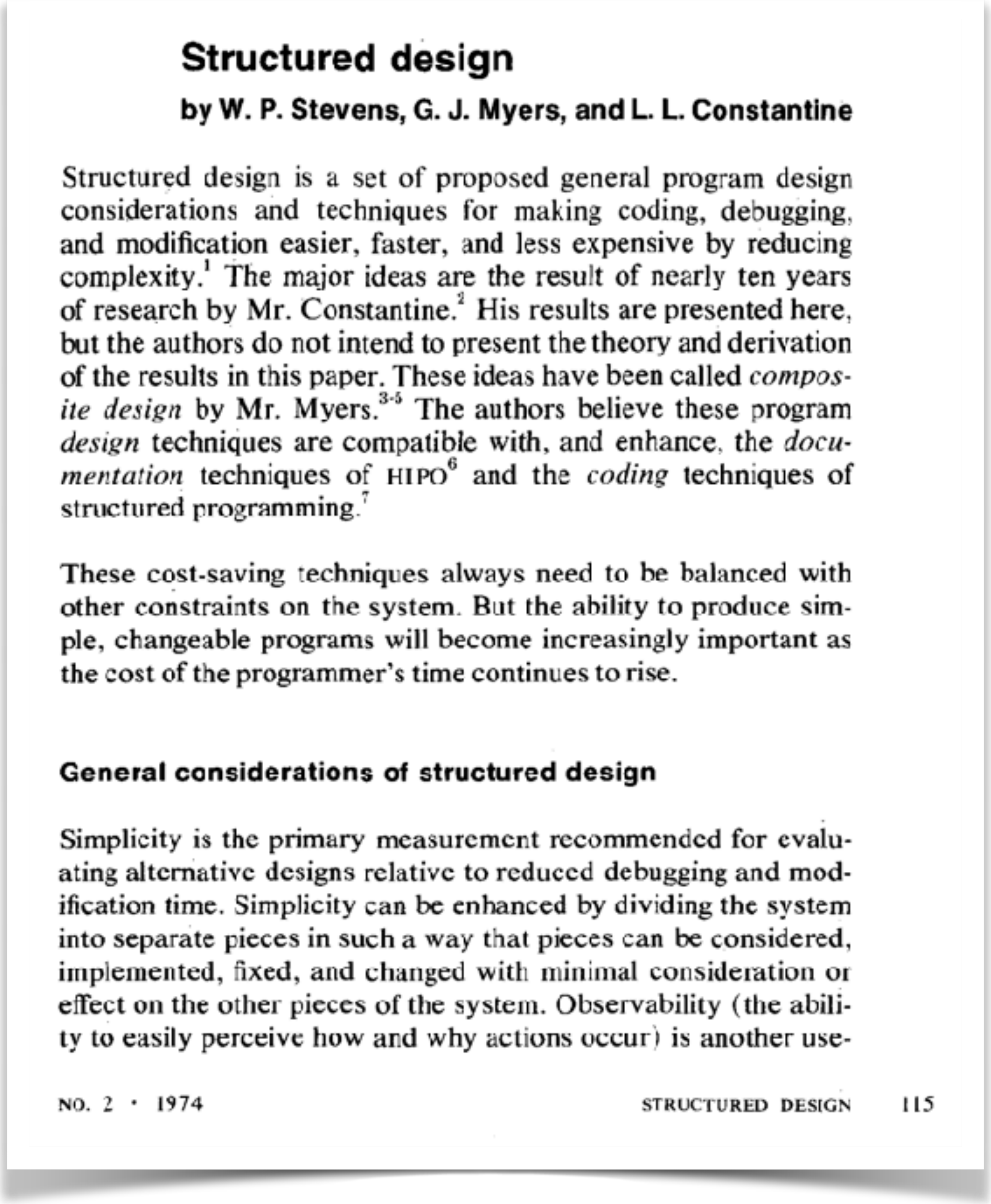
Logical on Coincidential Cohesion are weaker than all of the others

Higher levels of binding often include characteristics of the lower ones and add relationships

Wayne P. Stevens, Glenford J. Myers, Larry L. Constantine, **Structured Design**, IBM Systems Journal, 13(2):115-139, 1974.

How do we achieve Functional Cohesion?

„A useful technique in determining whether a module is functionally bound is writing a sentence describing the function(purpose) of the module, and then examining the sentence. “



Structured design

by W. P. Stevens, G. J. Myers, and L. L. Constantine

Structured design is a set of proposed general program design considerations and techniques for making coding, debugging, and modification easier, faster, and less expensive by reducing complexity.[1] The major ideas are the result of nearly ten years of research by Mr. Constantine.[2] His results are presented here, but the authors do not intend to present the theory and derivation of the results in this paper. These ideas have been called *composite design* by Mr. Myers.[3-5] The authors believe these program *design* techniques are compatible with, and enhance, the *documentation* techniques of HIPO[6] and the *coding* techniques of structured programming.[7]

These cost-saving techniques always need to be balanced with other constraints on the system. But the ability to produce simple, changeable programs will become increasingly important as the cost of the programmer's time continues to rise.

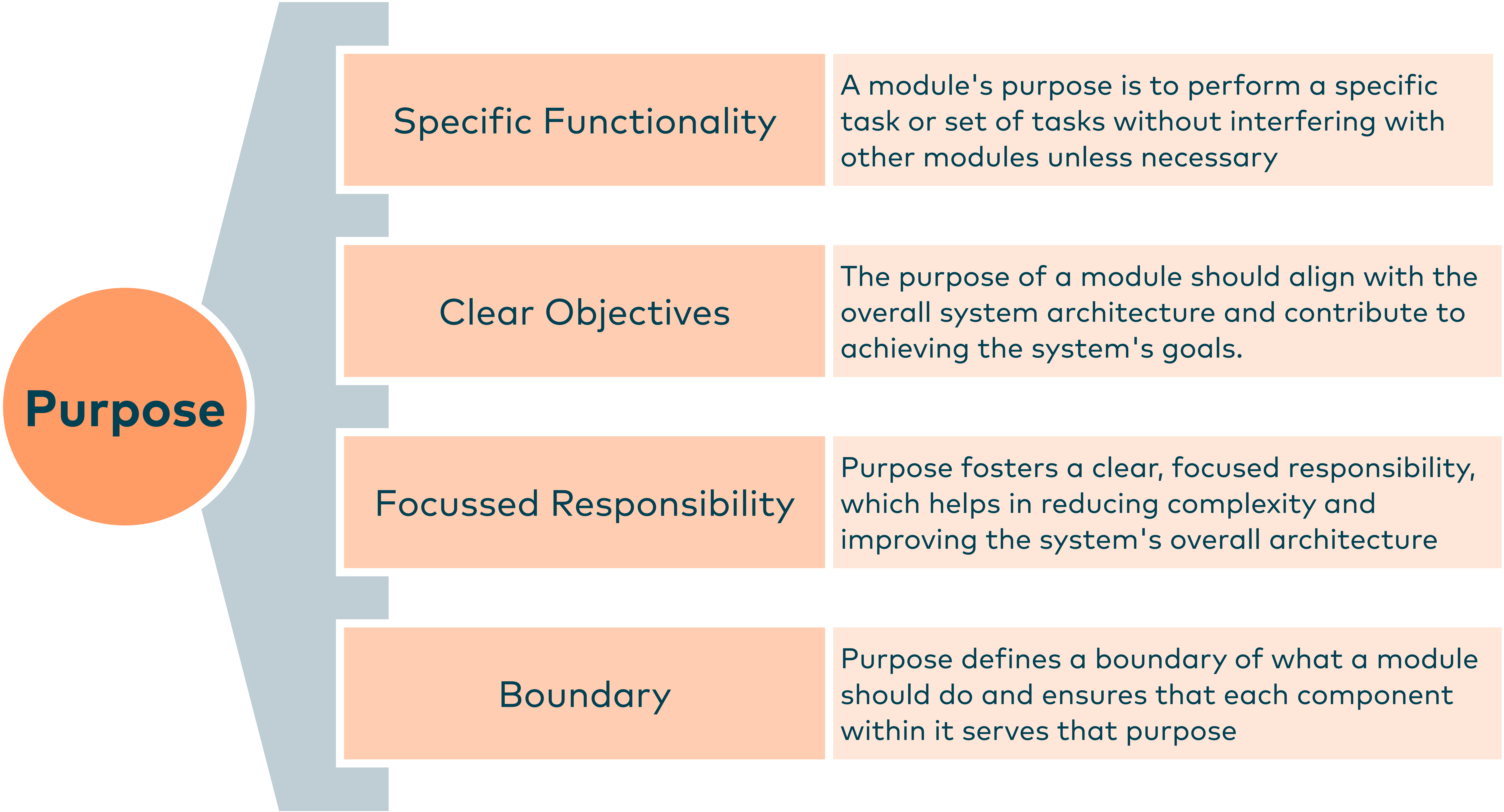
General considerations of structured design

Simplicity is the primary measurement recommended for evaluating alternative designs relative to reduced debugging and modification time. Simplicity can be enhanced by dividing the system into separate pieces in such a way that pieces can be considered, implemented, fixed, and changed with minimal consideration or effect on the other pieces of the system. Observability (the ability to easily perceive how and why actions occur) is another use-

NO. 2 · 1974 STRUCTURED DESIGN 115

Wayne P. Stevens, Glenford J. Myers, Larry L. Constantine, **Structured Design**, IBM Systems Journal, 13(2):115-139, 1974.

Cohesion and Purpose



Purpose

Specific Functionality

A module's purpose is to perform a specific task or set of tasks without interfering with other modules unless necessary

Clear Objectives

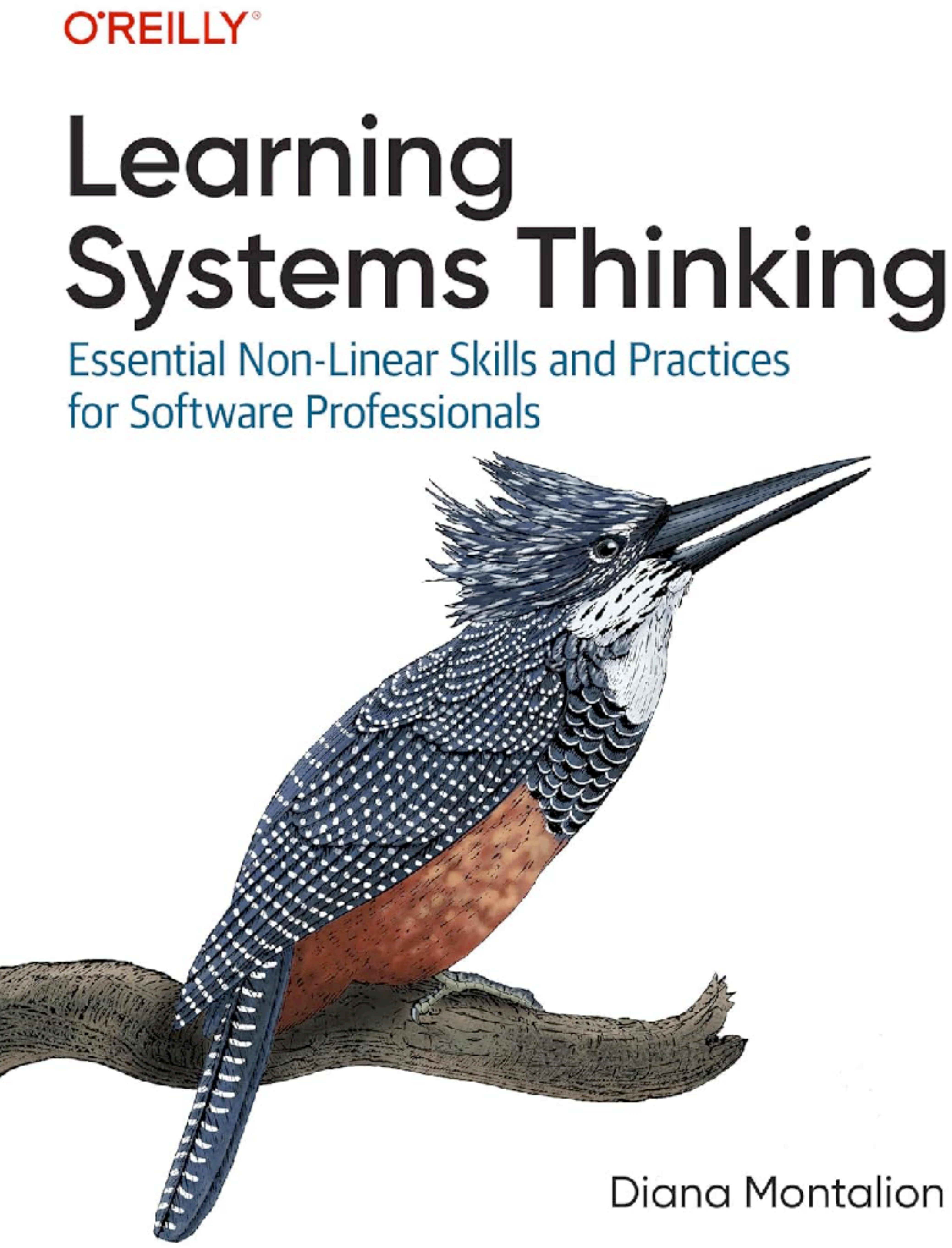
The purpose of a module should align with the overall system architecture and contribute to achieving the system's goals.

Focussed Responsibility

Purpose fosters a clear, focused responsibility, which helps in reducing complexity and improving the system's overall architecture

Boundary

Purpose defines a boundary of what a module should do and ensures that each component within it serves that purpose



O'REILLY

Learning Systems Thinking

Essential Non-Linear Skills and Practices for Software Professionals

Diana Montalion

Before diving into solutionizing, define the problem and why it's valuable to solve.

Problem Space

WHAT

is the challenge we need to solve?

Solution Space

HOW

do we solve the challenge?

Problem Space

WHAT

is the challenge we need to solve?

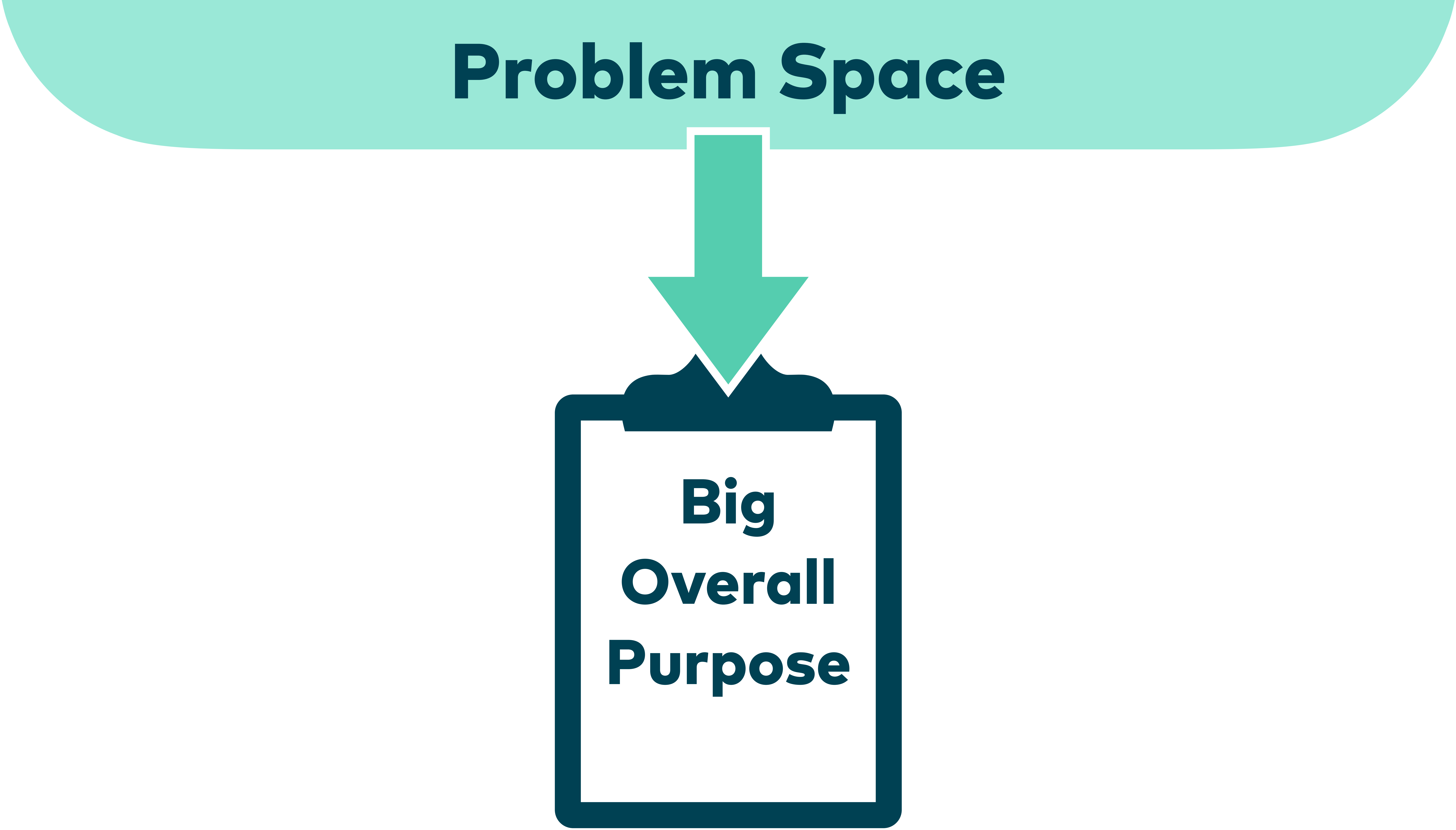
Value Proposition

Business Model

Business Challenge

Environment

User Needs



Problem Space

Big Overall Purpose

This is the starting point, we need to break this down to smaller parts in order to manage the complexity of a system

Problem Space

WHAT

is the challenge we
need to solve?

BIG COMPLEX

Challenge

Manageable Challenge

Defines

Purpose

Align

Module

Manageable Challenge

Defines

Purpose

Align

Module

Manageable Challenge

Defines

Purpose

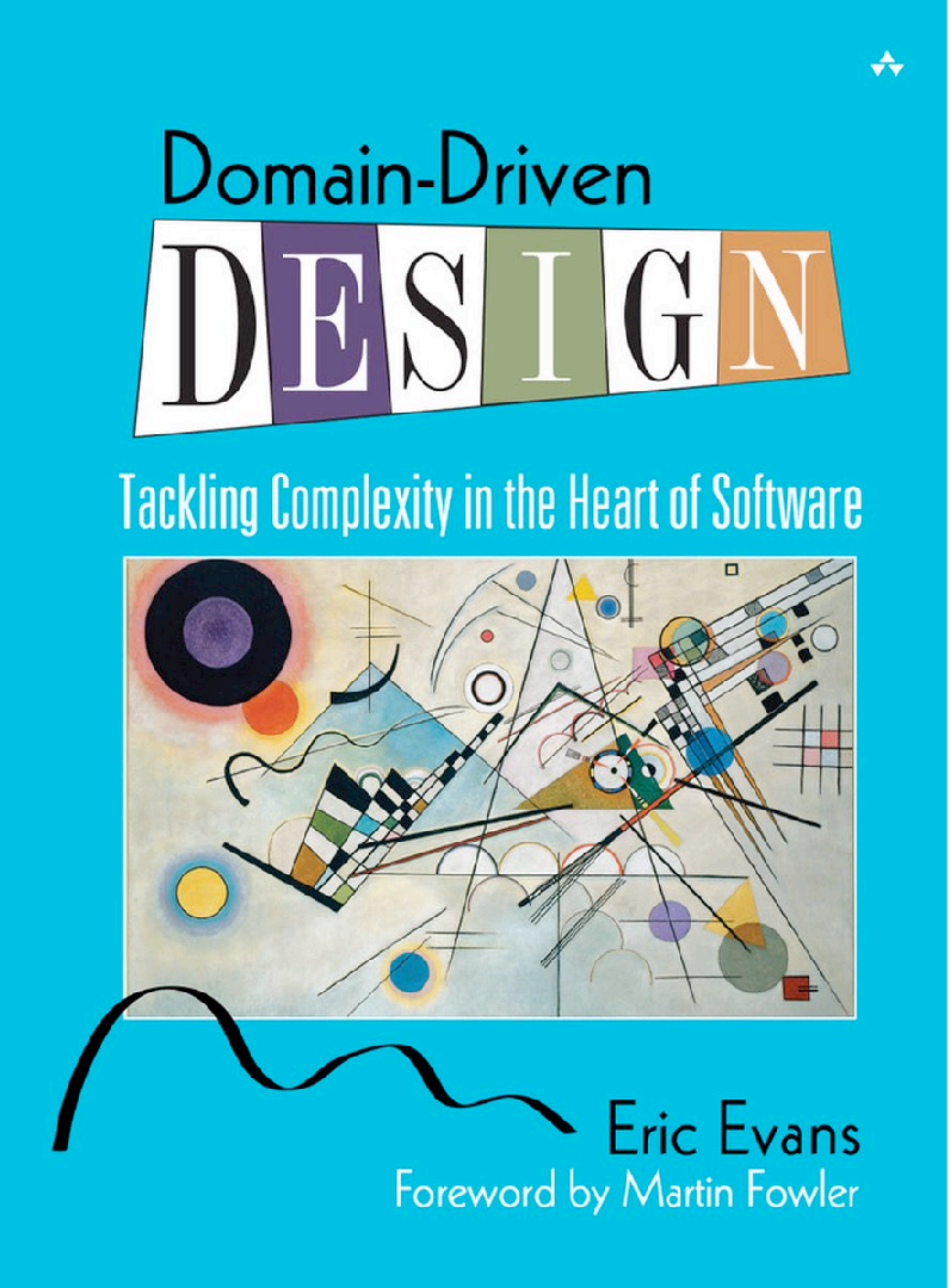
Align

Module

Purpose & Cohesion

in

Domain Driven Design


Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler

A model in a Bounded Context should have high cohesion, meaning that it forms a coherent whole around a single, unified purpose.

Problem Space

WHAT

is the challenge we
need to solve?

DOMAIN

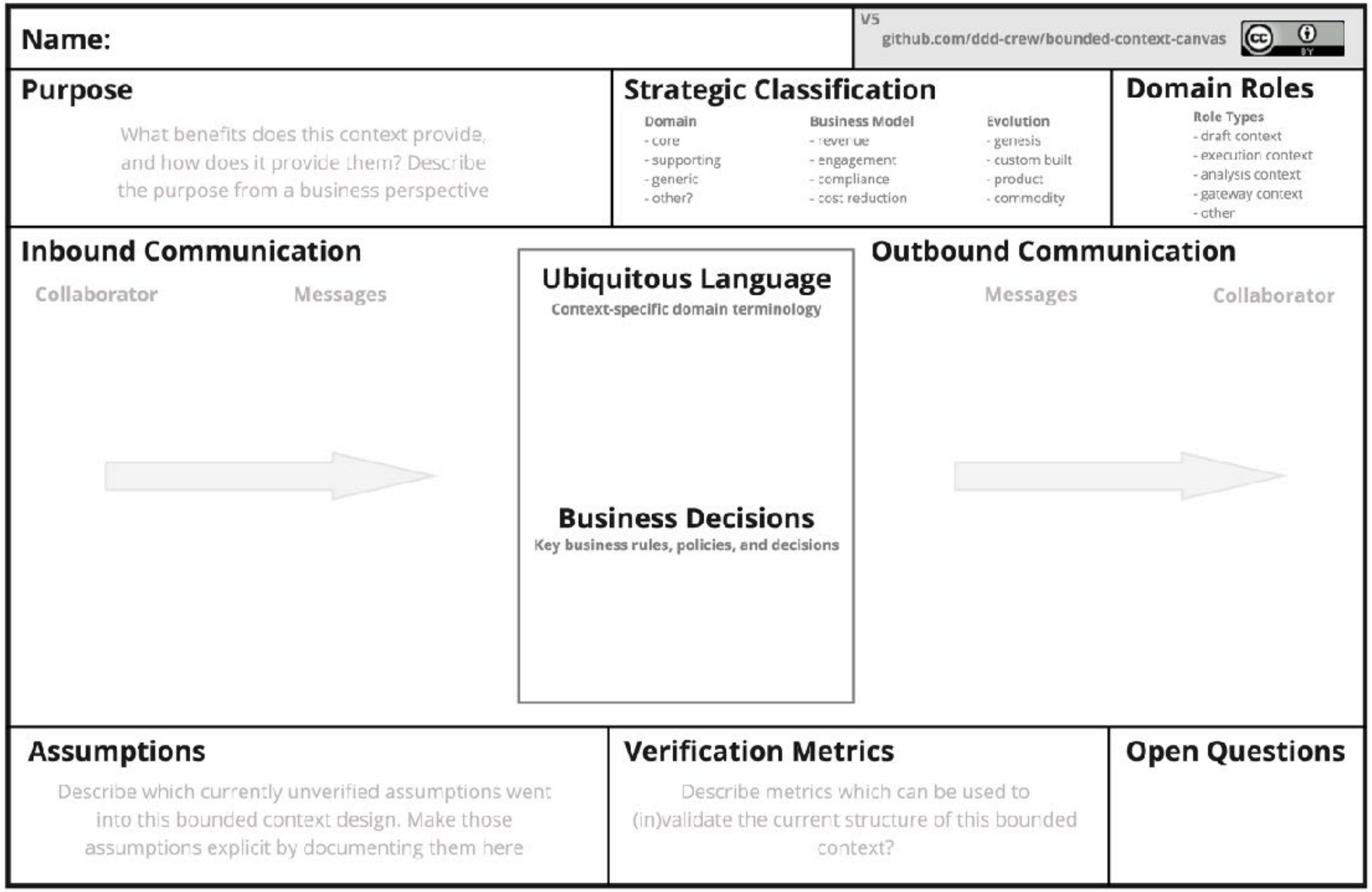
Subdomain → Defines → Purpose ← Solve ← Bounded Context

Subdomain → Defines → Purpose ← Solve ← Bounded Context

Subdomain → Defines → Purpose ← Solve ← Bounded Context

Defines
Purpose
Solve
Bounded Context
Align
Software Module
Defines
Purpose
Solve
Bounded Context
Align
Software Module
Defines
Purpose
Solve
Bounded Context
Align
Software Module

Bounded Context Design Canvas



Name:

V5
github.com/ddd-crew/bounded-context-canvas

Purpose

What benefits does this context provide, and how does it provide them? Describe the purpose from a business perspective

Strategic Classification

Domain
- core
- supporting
- generic
- other?

Business Model
- revenue
- engagement
- compliance
- cost reduction

Evolution
- genesis
- custom built
- product
- commodity

Domain Roles

Role Types
- draft context
- execution context
- analysis context
- gateway context
- other

Inbound Communication

Collaborator

Messages

Ubiquitous Language

Context-specific domain terminology

Business Decisions

Key business rules, policies, and decisions

Outbound Communication

Messages

Collaborator

Assumptions

Describe which currently unverified assumptions went into this bounded context design. Make those assumptions explicit by documenting them here

Verification Metrics

Describe metrics which can be used to (in)validate the current structure of this bounded context?

Open Questions

Source: https://github.com/ddd-crew/bounded-context-canvas

Bounded Context Design Canvas

Name:

Purpose

What benefits does this context provide, and how does it provide them? Describe the purpose from a business perspective

The Structured Design paper describes heuristics for determining the level of cohesion based on the textual description of the purpose.

Wayne P. Stevens, Glenford J. Myers, Larry L. Constantine, **Structured Design**, IBM Systems Journal, 13(2):115-139, 1974.

Sequential or communicational Cohesion

If you check at least one of the following criteria:

◉ Sentence has to be a compound sentence

◉ Contains a comma

◉ Contains more than one verb

Then the module is probably performing more than one function

Therefore, it probably has sequential or communicational binding

Wayne P. Stevens, Glenford J. Myers, Larry L. Constantine, **Structured Design**, IBM Systems Journal, 13(2):115-139, 1974.

Sequential or temporal Cohesion

If the sentence contains words relating to time such as:

- First
- Next
- Then
- After
- When
- Start

Then the module probably has sequential or temporal binding

Wayne P. Stevens, Glenford J. Myers, Larry L. Constantine, **Structured Design**, IBM Systems Journal, 13(2):115-139, 1974.

Temporal Cohesion

If the sentence contains words such as:

- Initialize
- Clean-Up
- Ramp-Up
- Tear-Down

Then the module probably has temporal binding

Wayne P. Stevens, Glenford J. Myers, Larry L. Constantine, **Structured Design**, IBM Systems Journal, 13(2):115-139, 1974.

Functional Cohesion

If the predicate of the sentence doesn't contain a single specific object following the verb

Then the module is probably functional bound

Example:

- Edit All Data implies logical binding
- Edit Credit Application may imply functional binding

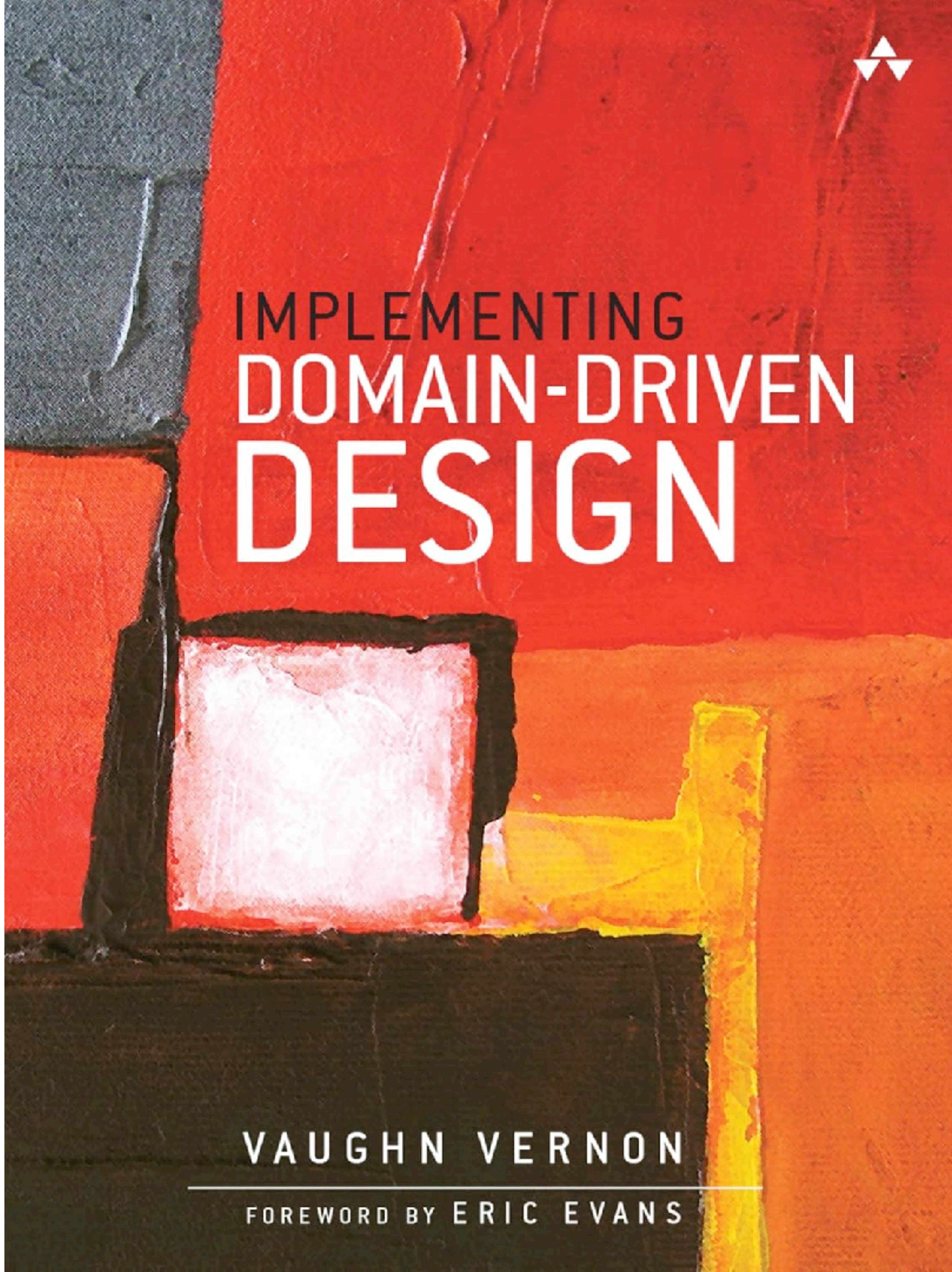
Wayne P. Stevens, Glenford J. Myers, Larry L. Constantine, **Structured Design**, IBM Systems Journal, 13(2):115-139, 1974.

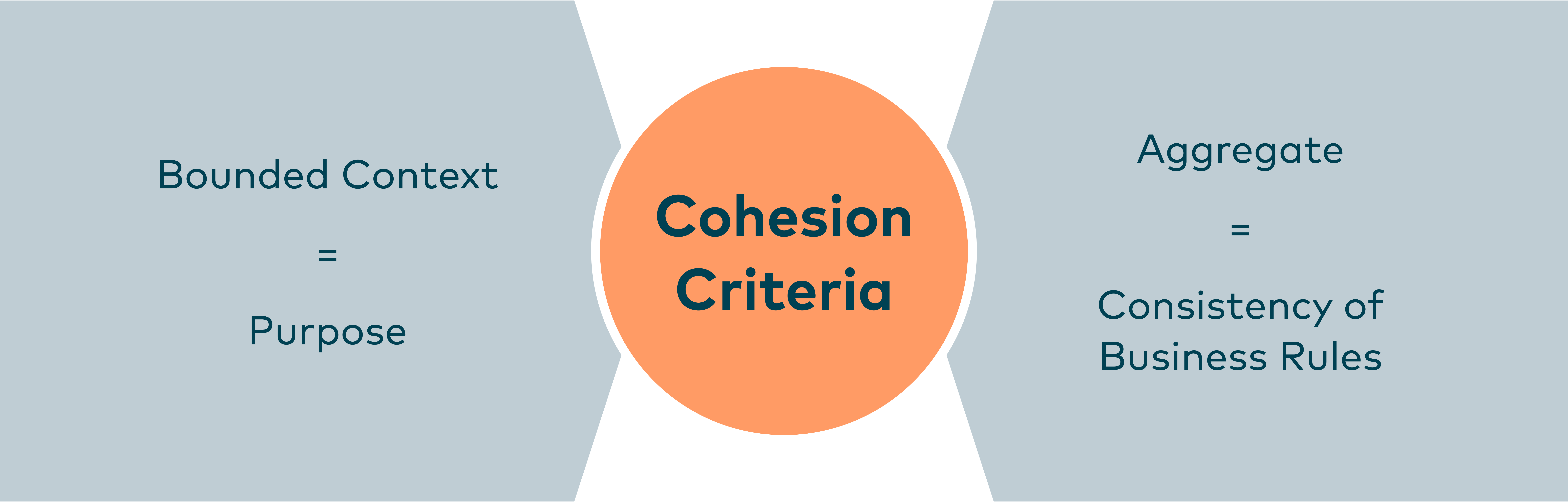
High cohesion is essential to **Aggregate** *design. Each Aggregate should fully encapsulate a business concept, keeping its responsibilities focused and manageable.*

…

When trying to discover the Aggregates in a Bounded Context we must understand the model's true invariants.

An invariant is a **business rule that must always be consistent**.





Bounded Context

=

Purpose

Cohesion Criteria

Aggregate

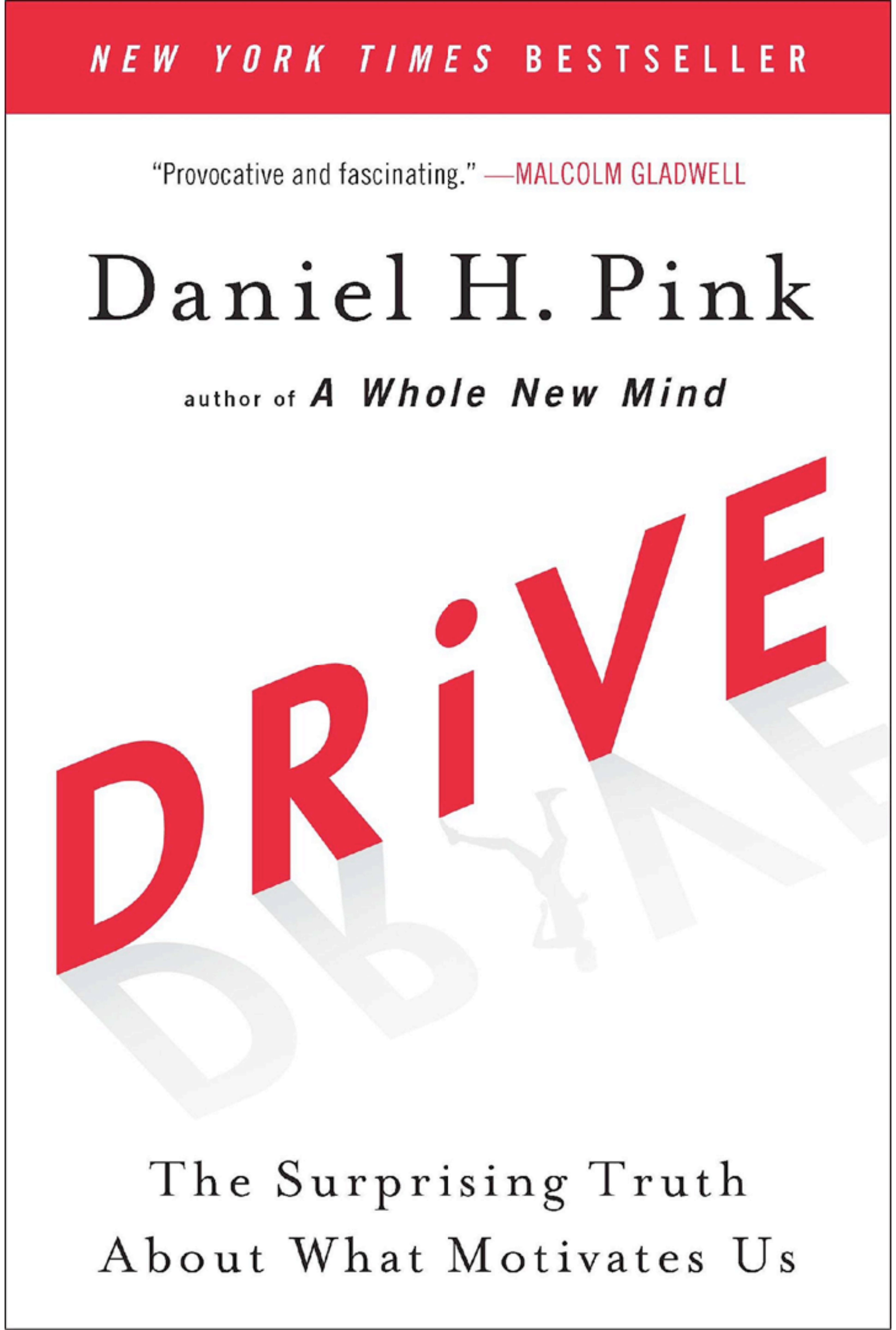
=

Consistency of Business Rules

Purpose

in

Teams & Organizations



NEW YORK TIMES BESTSELLER

"Provocative and fascinating." —MALCOLM GLADWELL

Daniel H. Pink

author of *A Whole New Mind*

DRIVE

The Surprising Truth About What Motivates Us

Autonomy

Desire to direct one's own life and refers to the freedom to choose how, when, and what tasks to perform.

Mastery

Desire to improve continuously and become better at something.

Purpose

Motivation to do work that aligns with a greater cause or mission, beyond just personal gain.

Autonomy

Desire to direct one’s own life and refers to the freedom to choose how, when, and what tasks to perform.

Teams are loosely coupled in organizational terms, cross-team coordination is greatly reduced.

Mastery

Desire to improve continuously and become better at something.

The members of a team can master the domain complexity of their area of responsibility and get better at it.

Purpose

Motivation to do work that aligns with a greater cause or mission, beyond just personal gain.

The area of responsibility of a team has a designated and distinct functional purpose.

We need (module) boundaries in software which enable teams to achieve these ideas

Autonomy

Teams are loosely coupled in organizational terms, cross-team coordination is greatly reduced.

Mastery

The members of a team can master the domain complexity of their area of responsibility.

Purpose

The area of responsibility of a team has a designated and distinct functional purpose.

Domain or Business Architecture

Defines

Purpose

Align

Software Architecture

Align

Team Organization

Cohesion

in

Modeling Work

Remember?

Software Architecture:

Structural Cohesion

Work in organization:

Social / linguistic Cohesion

Sociocultural Anthropology

Strength of cultural, social, or kinship ties that bind a community or group together. Social cohesion in traditional societies often stems from shared customs, rituals, and kinship structures that maintain group identity and solidarity.

Soil Mechanics

Attraction between particles within a soil, typically due to moisture or the presence of clays, which can hold the particles together. Cohesion helps determine the stability of soils and their ability to form aggregates, affecting erosion resistance and soil strength.

Chemistry

Force of attraction between molecules of the same substance. This is related to intermolecular forces, such as hydrogen bonds in water or van der Waals forces in non-polar molecules.

Civil Engineering

Ability of construction materials, like concrete or soil, to hold together. The cohesive forces in these materials contribute to their overall structural stability.

Political Science

In political science, cohesion often refers to the unity and agreement within political parties or factions. It is related to how consistently members of a party vote together or act in concert to achieve their political goals.

Linguistics

Ways in which different parts of a text or discourse are connected to create meaning. It includes grammatical and lexical linking, such as pronouns, conjunctions, and repetitions, which help to unify a text.

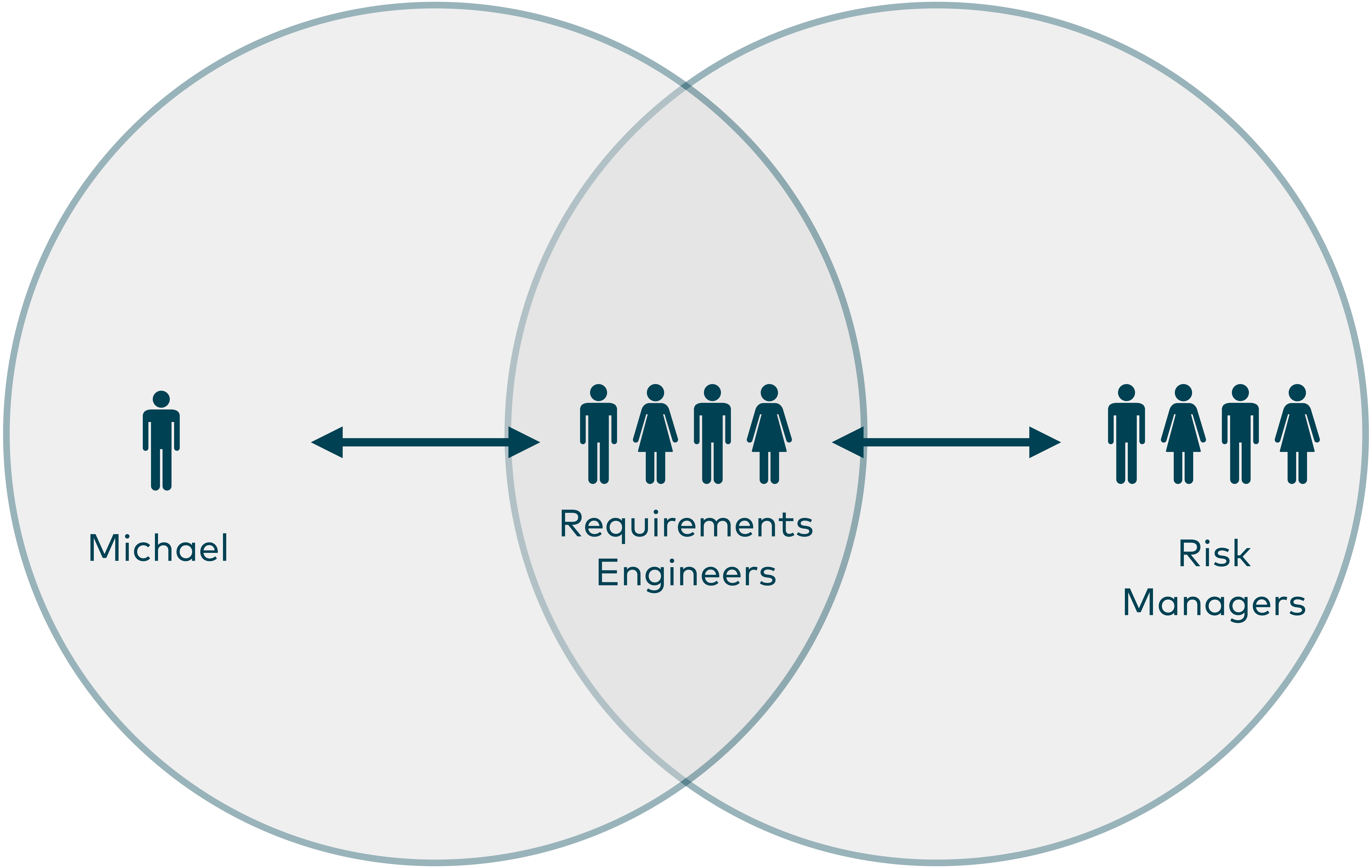
„It is not the domain experts knowledge that goes into production, it is the assumption of the developers that goes into production"

Alberto Brandolini

Erfinder des EventStormings

Let me tell you a story

Michael as a young developer for a mortgage loan scoring engine


Michael
Requirements Engineers
Risk Managers

I had a great and thorough specification

Let’s look at the business rules

Existing customers with an amount of cash of more than 10.000 EUR on accounts of the Pug Bank will be preferred per person with more points (+ 5 points).

Customers with equity ratio of 15 - 20 percent get 5 points, customers with equity ratio > 20 percent get 10 points and customers with > 30 percent get 15 points

> 3 warnings in the credit agency response are a nogo criteria

Total amount of loans > collateral value is a nogo criteria

Probability of repayment from credit agency query = points

Is the market value of the property in the average range that gives 10 points

< 120 points: red

A negative remark in the result of the credit agency query is a nogo criteria

Applicants from Munich and Hamburg will be preferred with more points (+ 5 points)

Probability of repayment from credit agency query < 60 is a nogo criteria

>= 120 points: green

A monthly budget surplus with future repayments of > 1,500 EUR gives 10 points.

Total amount of loans + sum of own funds != Sum of (purchase) costs is a nogo criteria

One nogo-criteria present: red (independent of amount of points)

Total monthly loan payments > monthly earning capacity - monthly expenses is a nogo criteria

My perspective on cohesion:

Points

A monthly budget surplus with future repayments of > 1,500 EUR gives 10 points.

Probability of repayment from credit agency query = points

Customers with equity ratio of 15 - 20 percent get 5 points, customers with equity ratio > 20 percent get 10 points and customers with > 30 percent get 15 points

Applicants from Munich and Hamburg will be preferred with more points (+ 5 points)

Is the market value of the property in the average range that gives 10 points

Existing customers with an amount of cash of more than 10.000 EUR on accounts of the Pug Bank will be preferred per person with more points (+ 5 points).

No Go Criteria

A negative remark in the result of the credit agency query is a nogo criteria

Total monthly loan payments > monthly earning capacity - monthly expenses is a nogo criteria

> 3 warnings in the credit agency response are a nogo criteria

Total amount of loans + sum of own funds != Sum of (purchase) costs is a nogo criteria

Probability of repayment from credit agency query < 60 is a nogo criteria

Total amount of loans > collateral value is a nogo criteria

Scoring Result

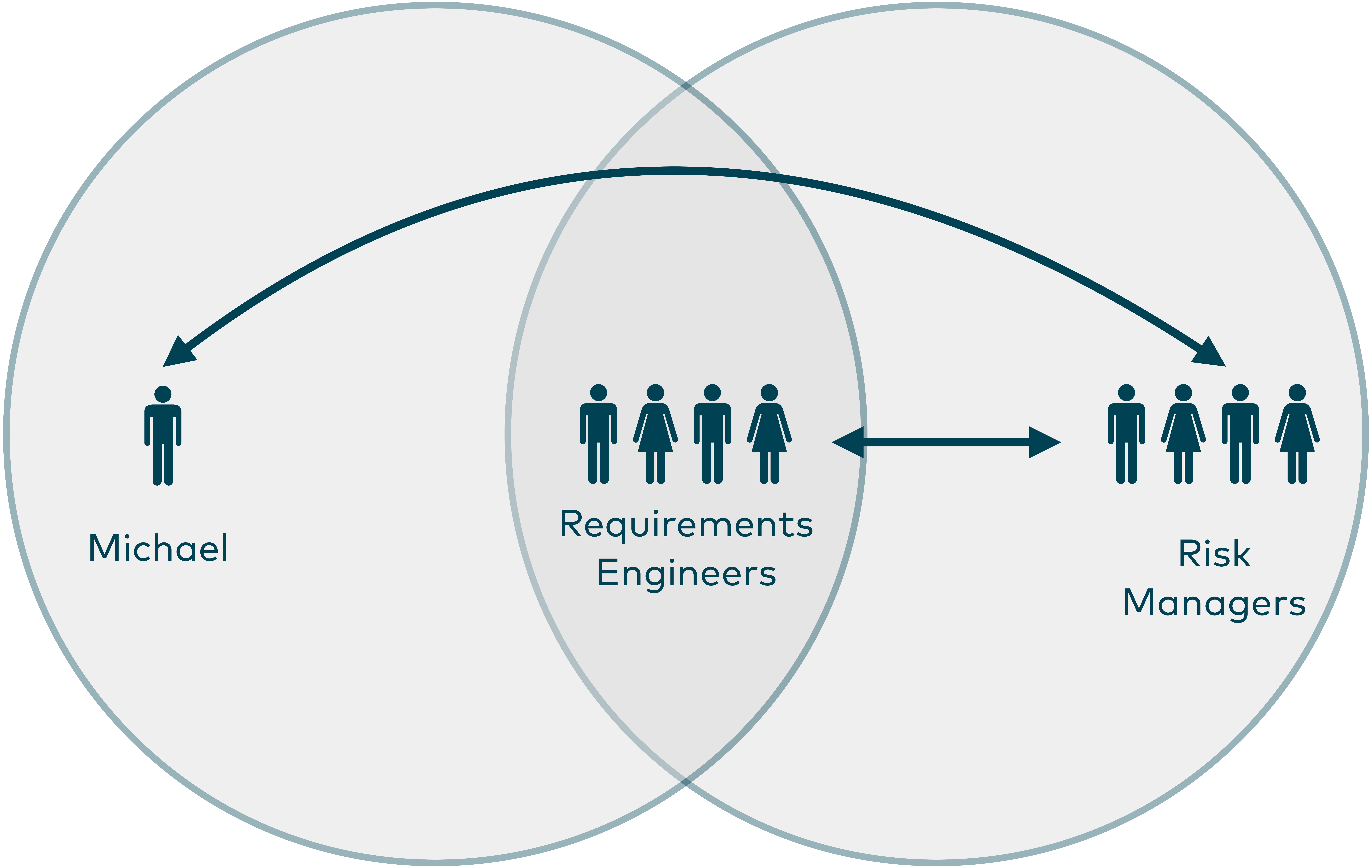
One nogo-criteria present: red (independent of amount of points)

< 120 points: red

>= 120 points: green

So I built my scoring engine according to that structure which I assumed in my head which roughly looked like this

The acceptance test and a change in communication


Michael
Requirements Engineers
Risk Managers

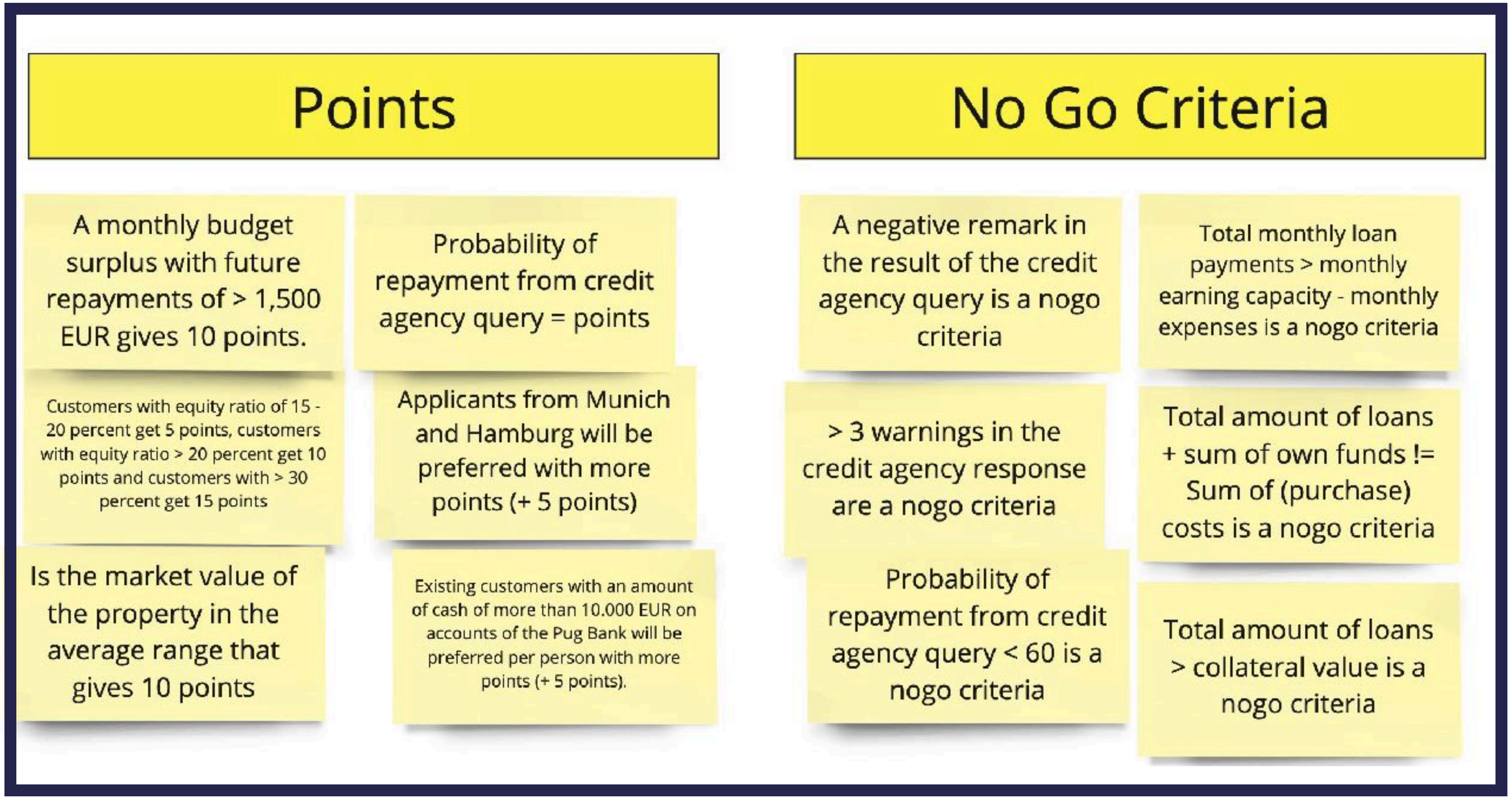
I developed a bad gut feeling but go-live was fine, just two minor bugs.

But then came a new requirement…

We want to see if a red scoring can become green with more own funds

if yes: how much more money do the applicants need?

Michael: „this affects nearly everything“



Points

A monthly budget surplus with future repayments of > 1,500 EUR gives 10 points.

Probability of repayment from credit agency query = points

Customers with equity ratio of 15 - 20 percent get 5 points, customers with equity ratio > 20 percent get 10 points and customers with > 30 percent get 15 points

Applicants from Munich and Hamburg will be preferred with more points (+ 5 points)

Is the market value of the property in the average range that gives 10 points

Existing customers with an amount of cash of more than 10.000 EUR on accounts of the Pug Bank will be preferred per person with more points (+ 5 points).

No Go Criteria

A negative remark in the result of the credit agency query is a nogo criteria

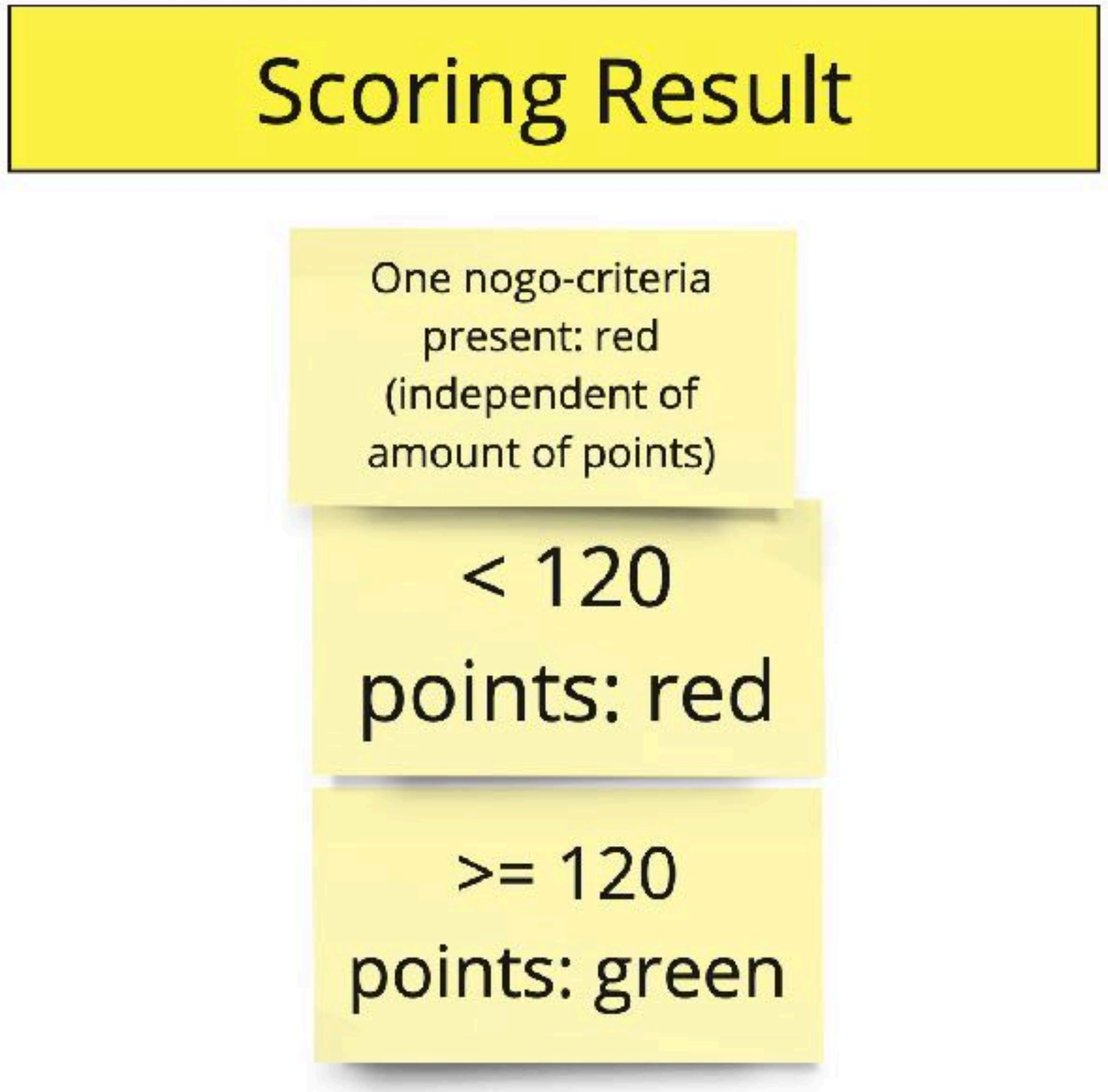
Total monthly loan payments > monthly earning capacity - monthly expenses is a nogo criteria

> 3 warnings in the credit agency response are a nogo criteria

Total amount of loans + sum of own funds != Sum of (purchase) costs is a nogo criteria

Probability of repayment from credit agency query < 60 is a nogo criteria

Total amount of loans > collateral value is a nogo criteria



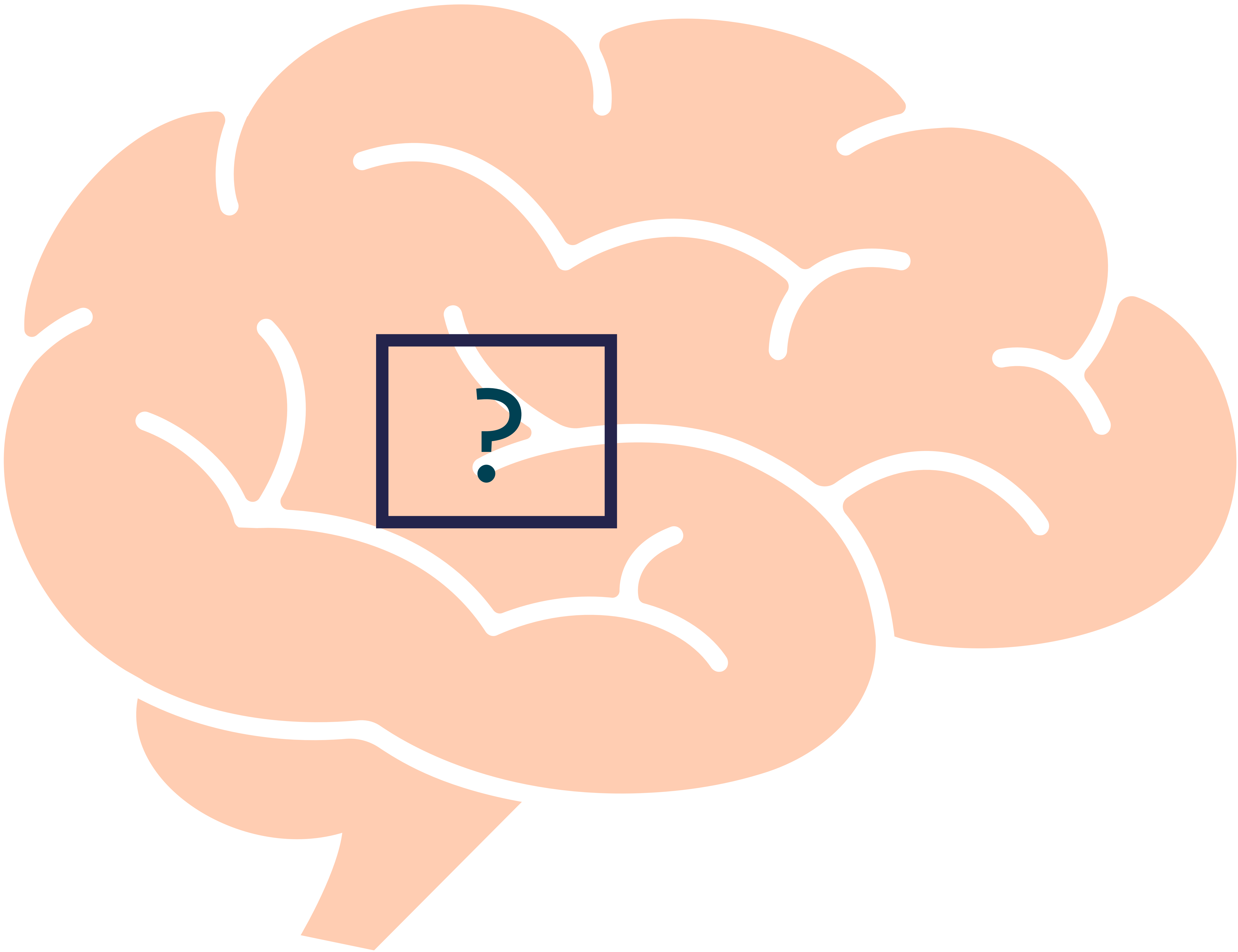
Scoring Result

One nogo-criteria present: red (independent of amount of points)

< 120 points: red

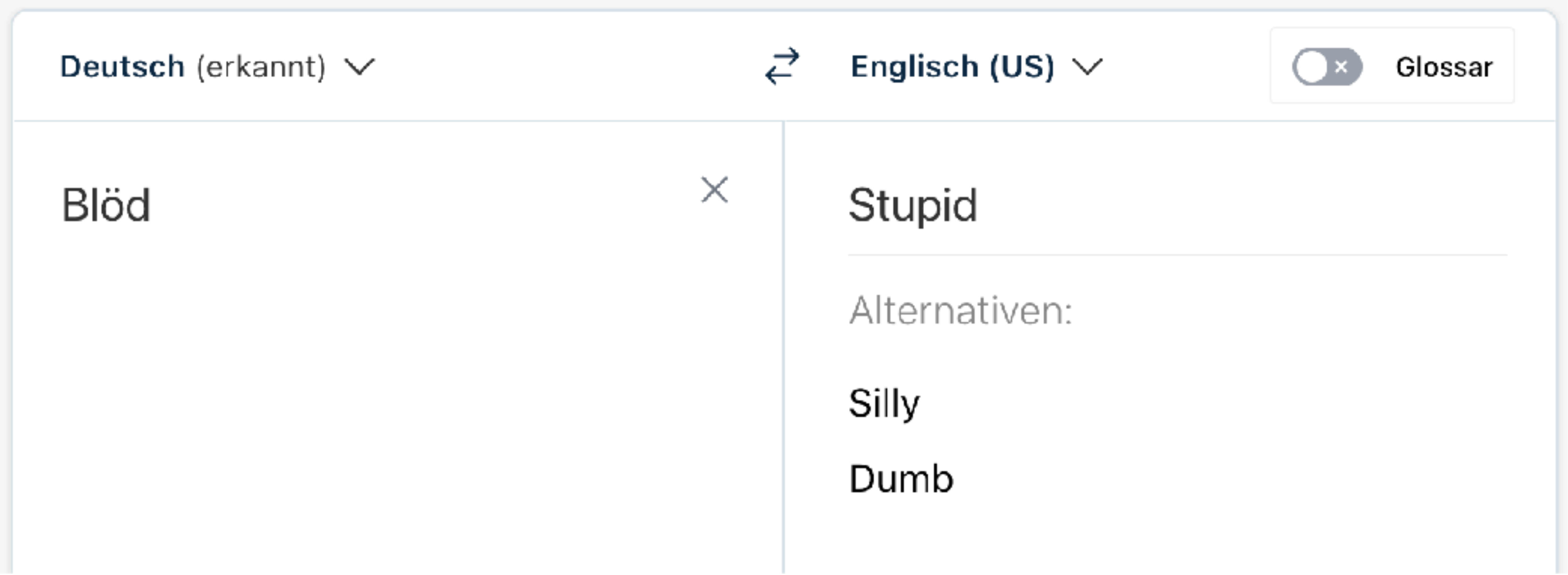
>= 120 points: green

Risk Management: „no, it's only in one part“



The risk managers started to think that I do my surname justice

(just replace the P in Plöd by a B)



Blöd

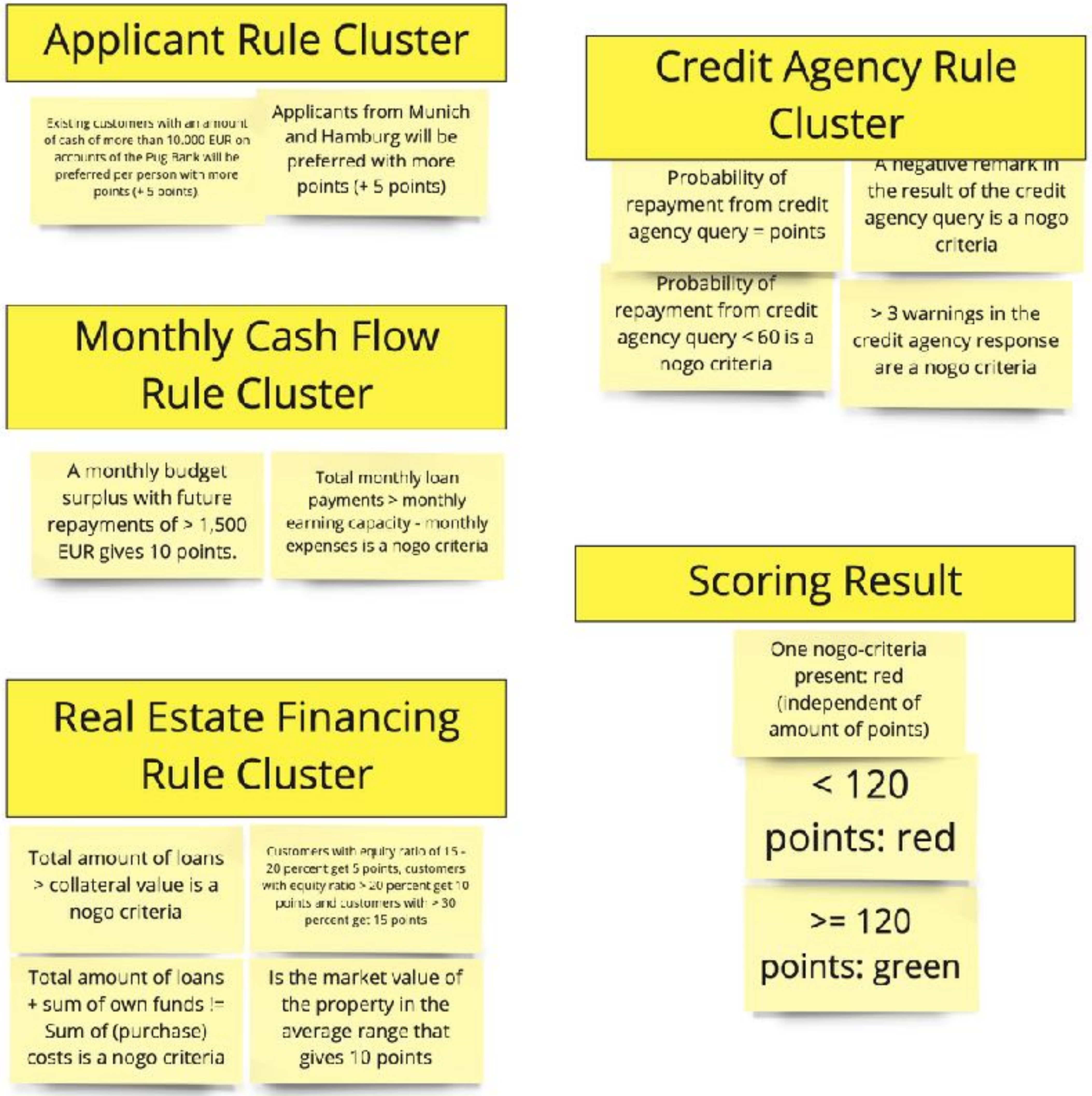
Stupid

Alternativen:

Silly

Dumb

Risk Management perspective on cohesion



Applicant Rule Cluster

Existing customers with an amount of cash of more than 10.000 EUR on accounts of the Pug Bank will be preferred per person with more points (+ 5 points)

Applicants from Munich and Hamburg will be preferred with more points (+ 5 points)

Monthly Cash Flow Rule Cluster

A monthly budget surplus with future repayments of > 1,500 EUR gives 10 points.

Total monthly loan payments > monthly earning capacity - monthly expenses is a nogo criteria

Real Estate Financing Rule Cluster

Total amount of loans > collateral value is a nogo criteria

Customers with equity ratio of 15 - 20 percent get 5 points, customers with equity ratio > 20 percent get 10 points and customers with > 30 percent get 15 points

Total amount of loans + sum of own funds != Sum of (purchase) costs is a nogo criteria

Is the market value of the property in the average range that gives 10 points

Credit Agency Rule Cluster

Probability of repayment from credit agency query = points

A negative remark in the result of the credit agency query is a nogo criteria

Probability of repayment from credit agency query < 60 is a nogo criteria

> 3 warnings in the credit agency response are a nogo criteria

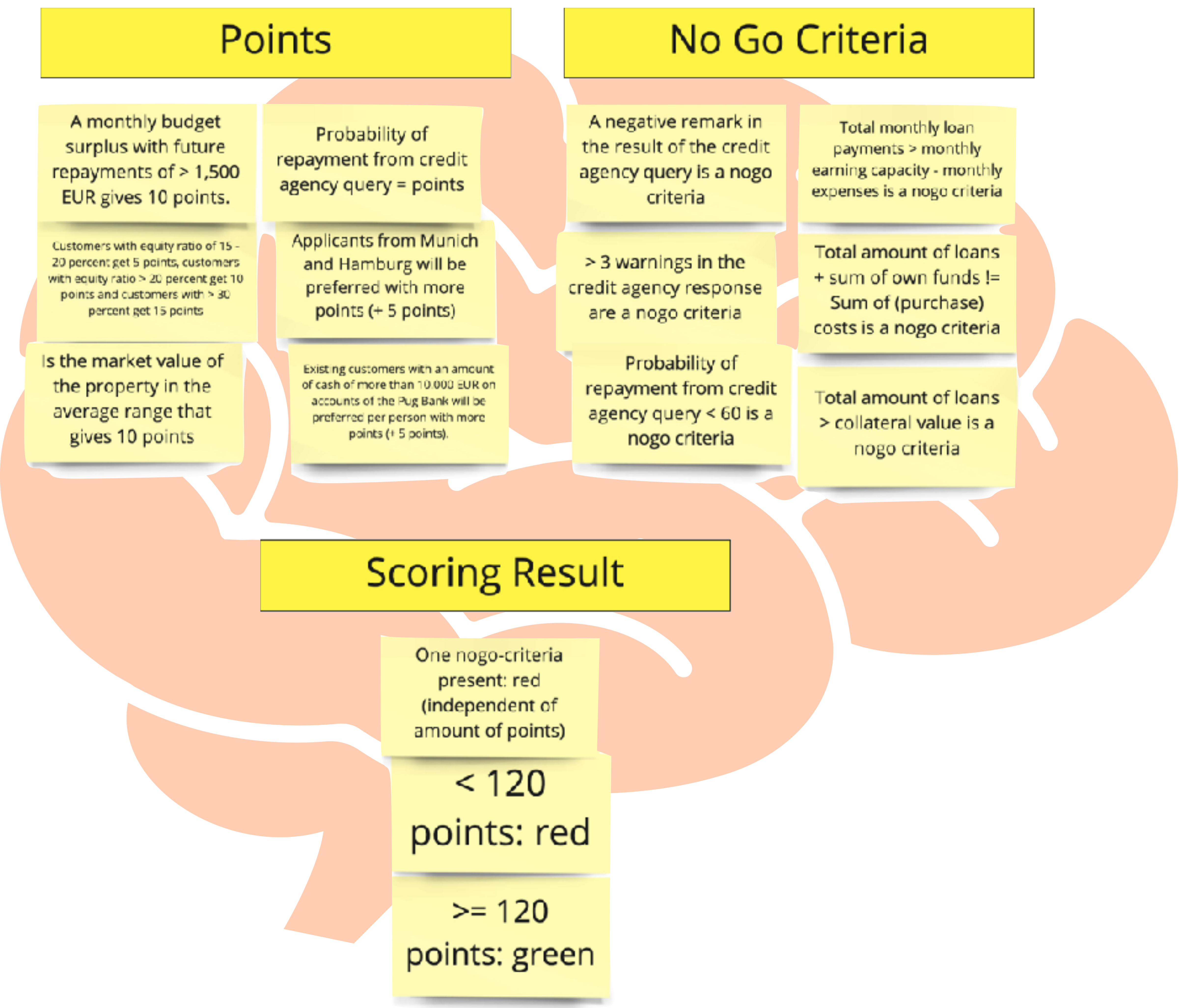
Scoring Result

One nogo-criteria present: red (independent of amount of points)

< 120 points: red

>= 120 points: green

We had different mental models on cohesiveness



Points

A monthly budget surplus with future repayments of > 1,500 EUR gives 10 points.

Probability of repayment from credit agency query = points

Customers with equity ratio of 15 - 20 percent get 5 points, customers with equity ratio > 20 percent get 10 points and customers with > 30 percent get 15 points

Applicants from Munich and Hamburg will be preferred with more points (+ 5 points)

Is the market value of the property in the average range that gives 10 points

Existing customers with an amount of cash of more than 10.000 EUR on accounts of the Pug Bank will be preferred per person with more points (+ 5 points).

No Go Criteria

A negative remark in the result of the credit agency query is a nogo criteria

Total monthly loan payments > monthly earning capacity - monthly expenses is a nogo criteria

> 3 warnings in the credit agency response are a nogo criteria

Total amount of loans + sum of own funds != Sum of (purchase) costs is a nogo criteria

Probability of repayment from credit agency query < 60 is a nogo criteria

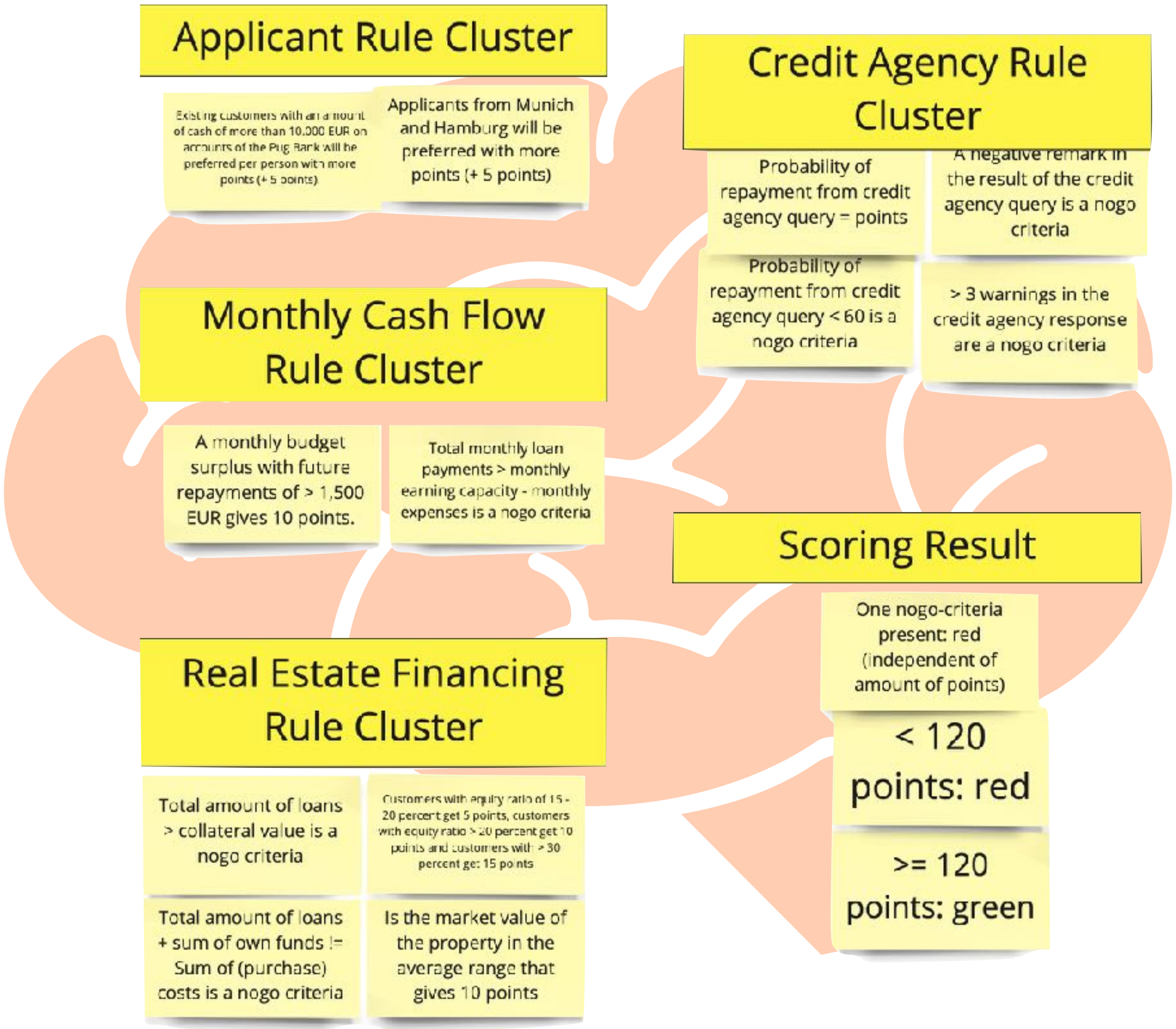
Total amount of loans > collateral value is a nogo criteria

Scoring Result

One nogo-criteria present: red (independent of amount of points)

< 120 points: red

>= 120 points: green



Applicant Rule Cluster

Existing customers with an amount of cash of more than 10.000 EUR on accounts of the Pug Bank will be preferred per person with more points (+ 5 points).

Applicants from Munich and Hamburg will be preferred with more points (+ 5 points)

Credit Agency Rule Cluster

Probability of repayment from credit agency query = points

A negative remark in the result of the credit agency query is a nogo criteria

Probability of repayment from credit agency query < 60 is a nogo criteria

> 3 warnings in the credit agency response are a nogo criteria

Monthly Cash Flow Rule Cluster

A monthly budget surplus with future repayments of > 1,500 EUR gives 10 points.

Total monthly loan payments > monthly earning capacity - monthly expenses is a nogo criteria

Scoring Result

One nogo-criteria present: red (independent of amount of points)

< 120 points: red

>= 120 points: green

Real Estate Financing Rule Cluster

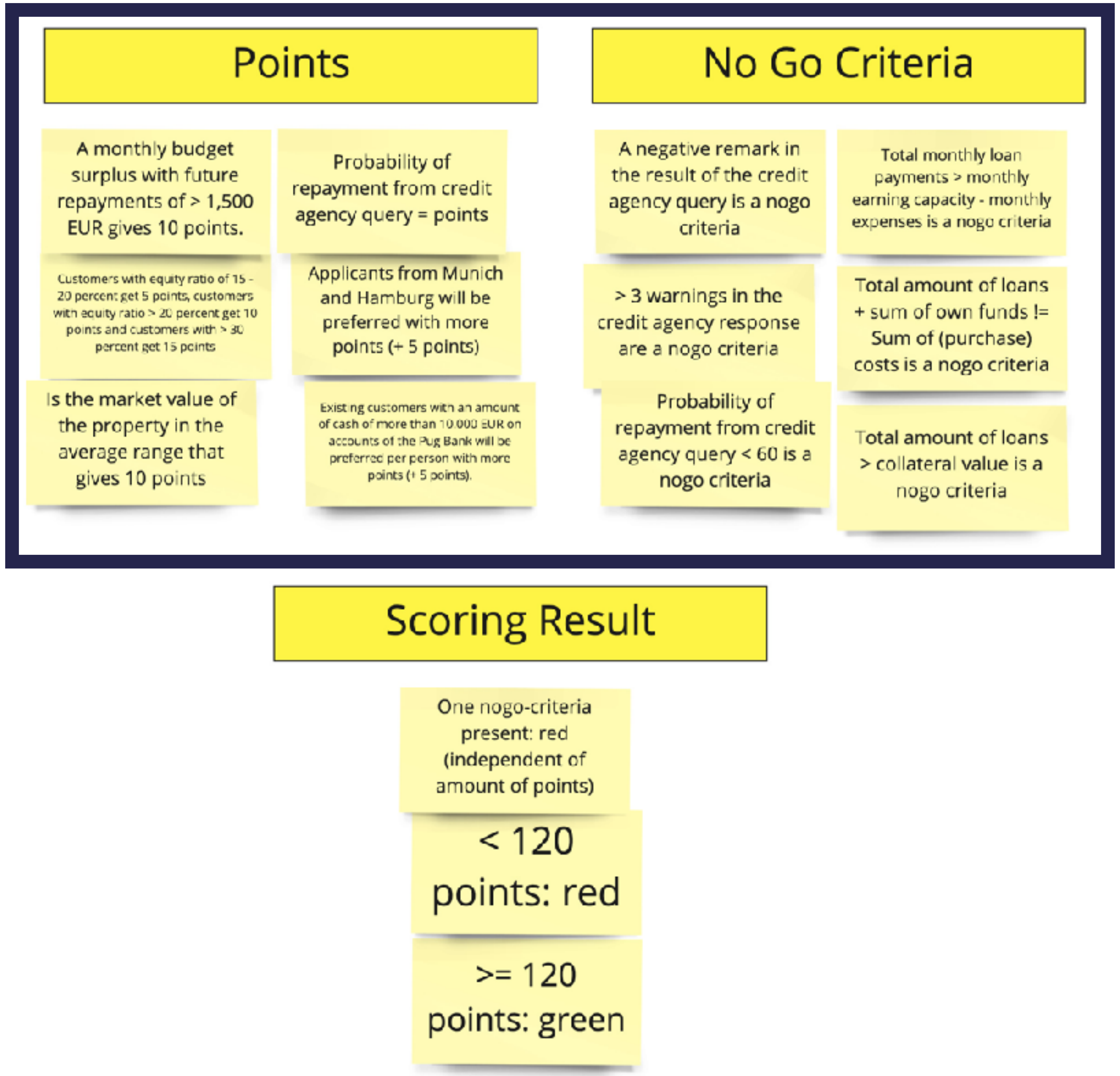
Total amount of loans > collateral value is a nogo criteria

Customers with equity ratio of 15 - 20 percent get 5 points, customers with equity ratio > 20 percent get 10 points and customers with > 30 percent get 15 points

Total amount of loans + sum of own funds != Sum of (purchase) costs is a nogo criteria

Is the market value of the property in the average range that gives 10 points

Everyone was right, from their perspective



Points

A monthly budget surplus with future repayments of > 1,500 EUR gives 10 points.

Probability of repayment from credit agency query = points

Customers with equity ratio of 15 - 20 percent get 5 points, customers with equity ratio > 20 percent get 10 points and customers with > 30 percent get 15 points

Applicants from Munich and Hamburg will be preferred with more points (+ 5 points)

Is the market value of the property in the average range that gives 10 points

Existing customers with an amount of cash of more than 10.000 EUR on accounts of the Pug Bank will be preferred per person with more points (+ 5 points).

No Go Criteria

A negative remark in the result of the credit agency query is a nogo criteria

Total monthly loan payments > monthly earning capacity - monthly expenses is a nogo criteria

> 3 warnings in the credit agency response are a nogo criteria

Total amount of loans + sum of own funds != Sum of (purchase) costs is a nogo criteria

Probability of repayment from credit agency query < 60 is a nogo criteria

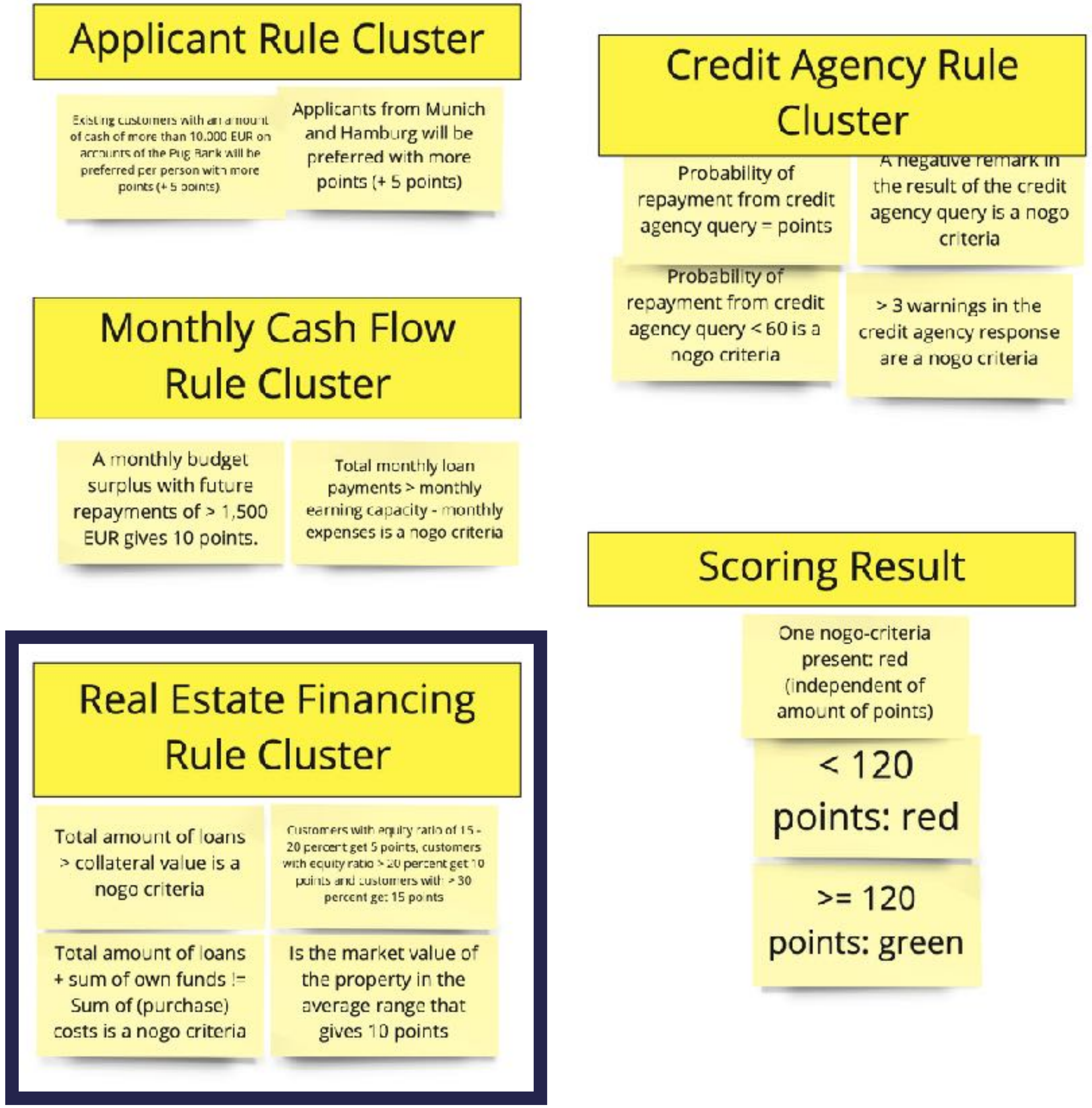
Total amount of loans > collateral value is a nogo criteria

Scoring Result

One nogo-criteria present: red (independent of amount of points)

< 120 points: red

>= 120 points: green



Applicant Rule Cluster

Existing customers with an amount of cash of more than 10.000 EUR on accounts of the Pug Bank will be preferred per person with more points (+ 5 points).

Applicants from Munich and Hamburg will be preferred with more points (+ 5 points)

Credit Agency Rule Cluster

Probability of repayment from credit agency query = points

A negative remark in the result of the credit agency query is a nogo criteria

Probability of repayment from credit agency query < 60 is a nogo criteria

> 3 warnings in the credit agency response are a nogo criteria

Monthly Cash Flow Rule Cluster

A monthly budget surplus with future repayments of > 1,500 EUR gives 10 points.

Total monthly loan payments > monthly earning capacity - monthly expenses is a nogo criteria

Real Estate Financing Rule Cluster

Total amount of loans > collateral value is a nogo criteria

Customers with equity ratio of 15 - 20 percent get 5 points, customers with equity ratio > 20 percent get 10 points and customers with > 30 percent get 15 points

Total amount of loans + sum of own funds != Sum of (purchase) costs is a nogo criteria

Is the market value of the property in the average range that gives 10 points

Scoring Result

One nogo-criteria present: red (independent of amount of points)

< 120 points: red

>= 120 points: green

I refactored my code to their mental model and the new requirement was suddenly very easy to implement

„the key to incremental architecture is to build on a framework that can accommodate change... that framework is the domain.... By modeling the domain, you can more easily handle changes to the domain“

Allen Holub

https://holub.com

If you only talk to your

„highly cohesive“ group of people with
„highly cohesive“ perspectives and
„highly cohesive“ biases and
„highly cohesive“ behaviors and
„highly cohesive“ attitudes

You run a fundamental risk of getting boundaries that

LACK FUNCTIONAL COHESION

The mental model that influences the perspective on attraction of concepts and models is heavily driven by the organization which embodies the social ties and customs of the people talking to each other during the design and modeling work.

THANK YOU!

INNOQ

www.innoq.com

Michael Plöd

E-Mail: michael.ploed@innoq.com
Socials: @bitboss (@mastodon.social / .bsky.social)
LinkedIn: https://www.linkedin.com/in/michael-ploed/

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim
+49 2173 3366-0

Ohlauer Str. 43
10999 Berlin

Ludwigstr. 180E
63067 Offenbach

Kreuzstr. 16
80331 München

Hermannstrasse 13
20095 Hamburg

Erftstr. 15-17
50672 Köln

Königstorgraben 11
90402 Nürnberg