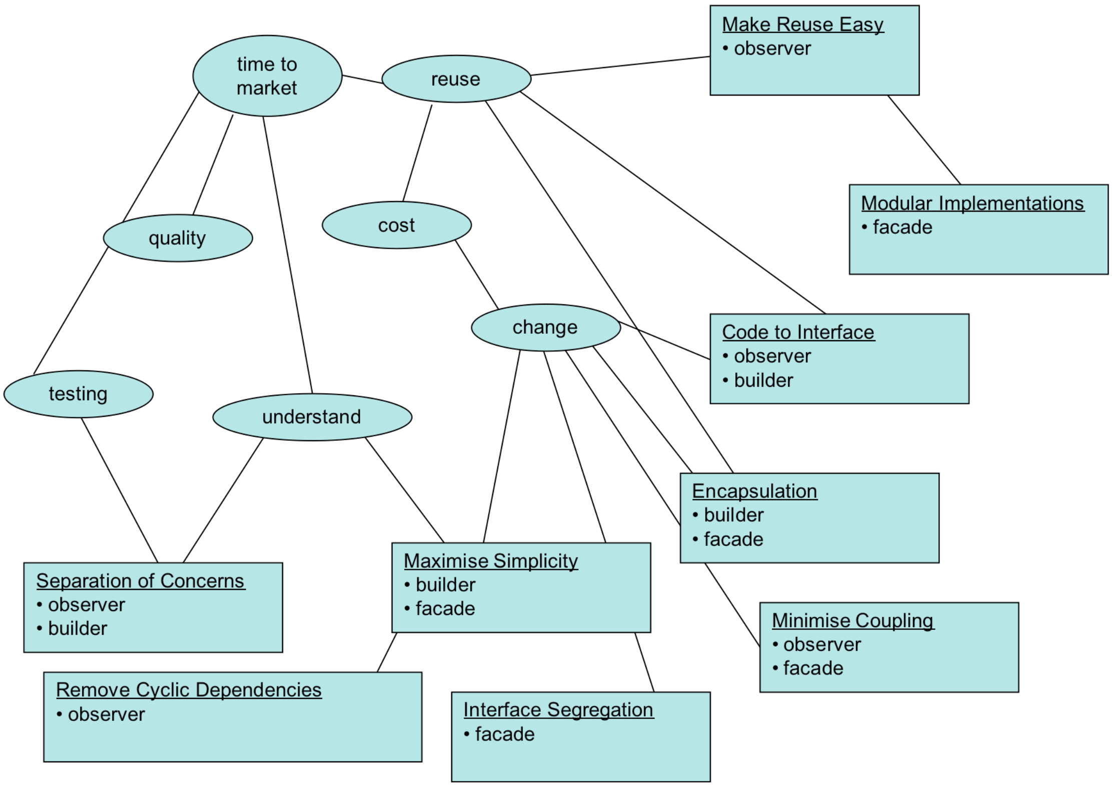
Software Design Principles – Mining Pattern DNA: Workshop Outputs

Group 1



Group 2

This group organised their outputs in a grid.
The horizontal axis, “Goals,” contained the following columns:

- maintainability
- optimise performance
- usability
- minimise storage
- minimise traffic

The vertical axis, “principles,” contained the following rows:

- maintainability
- encapsulation
- loose coupling
- don’t repeat yourself
- separation of concerns
- open / closed
- program to interfaces
- Liskov substitution principle
- dependency inversion
- consistent appearance on screen
- make reuse easy

Group 3

Group 3 arranged their principles in a rectangle whose four edges were marked (going North ->East->South->West round the rectangle):

- easy to use (now)
- easy to extend (future)
- easy to verify (test)
- easy to change (future)

They defined the following principles and associated patterns, and placed them in the rectangle.

Single Responsibility

easy to use / easy to extend

- builder
- layers
- chain of responsibility

Encapsulation

easy to extend

- facade
- strategy

Programming in the Client's Language

easy to use

- facade
- adapter

Cohesion

easy to change

- layers
- facade

Acyclic Dependencies

easy to change

- layers
- observer
- mediator

Encapsulate Variation

easy to change

- builder / factory
- command
- visitor / template

Abstractions Live Longer Than Details

easy to change / easy to verify

- layers

- observer
- command
- template

Dependency Inversion

easy to verify

- builder
- factory
- visitor

Minimise Coupling

all

- facade
- builder
- observer

Interface Segregation

all

- facade
- builder

Open / Closed

all

- builder
- abstract factory
- observer

Simplicity

all

- facade
- decorator

Code to Interfaces

all

- facade
- decorator