WP08

From Domains to Aspects

Design of Software Architectures

Dr. Vadim Zaytsev aka @grammarware, 20 September 2022

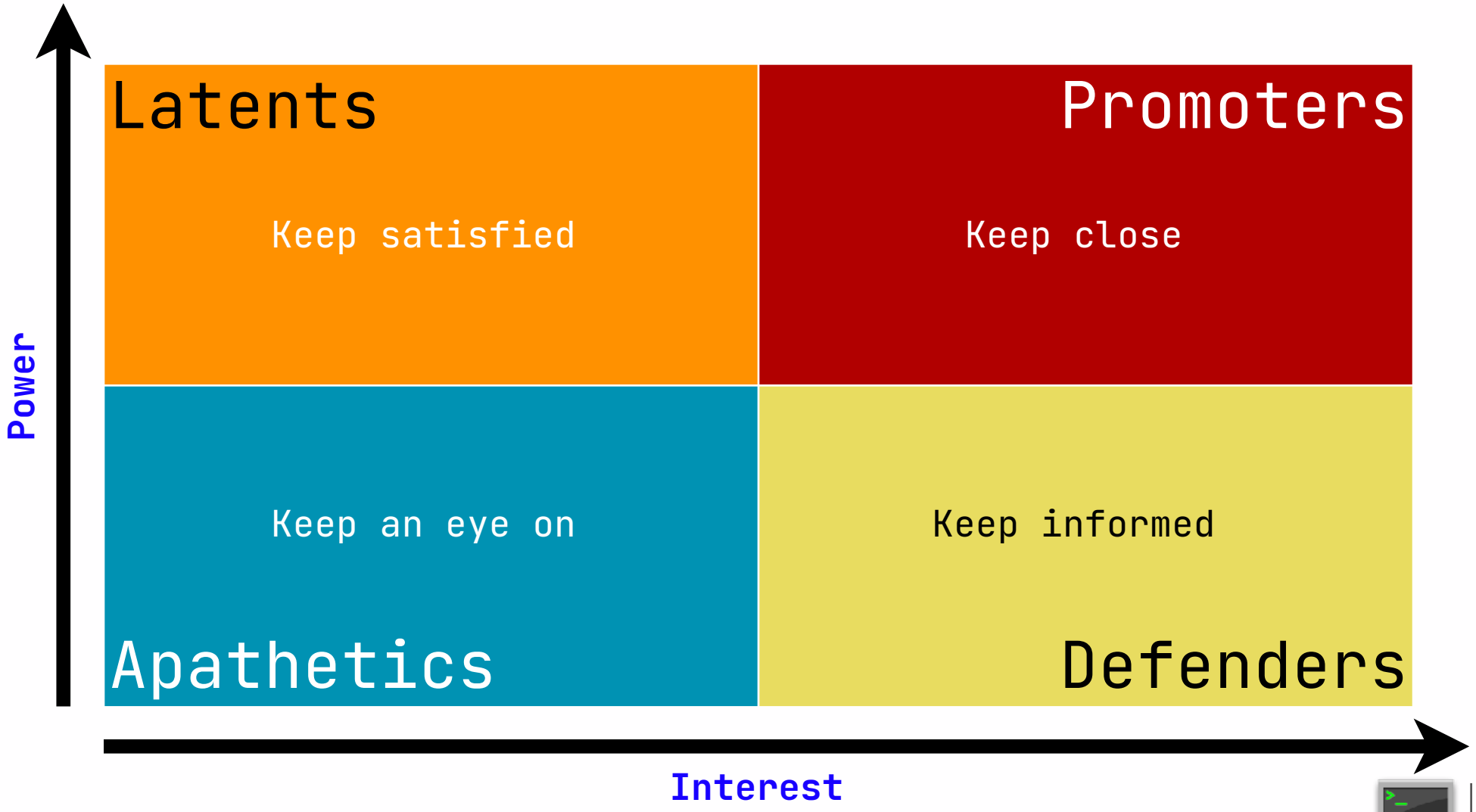
UNIVERSITY OF TWENTE.

Deadlines

- Intermediate feedback
  - ~~19 September 18:00 (WP07)~~
  - 12 October 12:00 (~WP15)
- Final document
  - 28 October 17:00 (~WP23)
- Individual
  - 11 November 17:00

week number	36	37	38	39	40	41	42	43	44	45
week type	M\L	M\L	M\L	M\L	M\L	M\L	M\L	M\L	M\E	MAE
quartile-week	1-01	1-02	1-03	1-04	1-05	1-06	1-07	1-08	1-09	1-10
Monday	5	12	D 19	26	3	10	17	24	31	7
Tuesday	6	13	S 20	27	4	D 11	18	25	1	8
Wednesday	S 7	S 14	21	S 28	5	S 12	S 19	26	2	9
Thursday	8	15	22	29	6	S 13	20	27	3	10
Friday	S 9	S 16	S 23	S 30	7	Information Day14	S 21	D 28	4	D 11
Saturday	10-09	17-09	24-09	01-10	08-10	15-10	22-10	29-10	05-11	12-11
Sunday	11-09	18-09	25-09	02-10	09-10	16-10	23-10	30-10	06-11	13-11

WP06 Recap: Stakeholder analysis



Role of the software architecture

to provide

solution direction

for

most important properties

that are

the most difficult to realise

More basic terminology

- **Domain**
  - area of coherent activity
- **Concept**
  - part of the domain jargon
- **Aspect**
  - the main concept of the architecture domain

Software architecture

- The software architecture of a system is
  - a collection of statements
  - that gives direction to
    - the design
    - the realisation and
    - the evolution
  - of the software in its environment
- Statements are a model of:
  - structure of the system elements and their relations, or
  - guidelines for creating structure, elements and their relationships

Statements are about aspects

- 1-1..*
- Aspects are coming from the domain
- The architect
  - thinks about aspects
  - communicates about aspects
  - reasons about aspects
  - is responsible for a set of aspects

Example aspect: performance

- Associated domains:
  - race cars / racing
  - software development
  - crypto mining
  - pro athletics
  - hardware
  - sport
  - hackers
  - production facilities
  - construction
  - business people like CEOs
- Related concepts:
  - efficiency
  - throughput
  - horsepower
  - measurements
  - torque
  - return of investment
  - speed
  - KPI

Example aspect: user experience

- Associated domains:
  - theatre
  - applications
  - websites
  - shop / mall
  - lessons
  - vending machine
  - traffic lights
  - travelling

- Related concepts:
  - interface
  - layout
  - feedback
  - perception
  - comfort
  - number of clicks
  - lo-fi/hi-fi prototype
  - interaction
  - fatigue
  - archetypes
  - delays
  - power user
  - persona

How to find aspects

- From experience:
  - ask a system expert or a domain expert
  - usual suspects in conversations about architecture
  - frameworks and checklists (e.g., IEEE 25010)
  - existing architecture [description]s
- From stakeholders:
  - interaction / functionality – things to do with the system
  - tasks – activities involving the system
  - concerns – things to worry about
- From observing the system:
  - things the system does or guards

Identify aspects in statements

TurboWorkbench allows companies to develop and express their own coding standards.

Loading is done by drones, so that ships do not have to alter their course and pay harbour fees.

COBOL-FIT always supports its customers in their own unique style of training.

A winning strategy in our game corresponds to a strategy for the robot to achieve a certain goal.

Aspects everywhere

Typical application aspects

- Legal compliance, actuality, traceability
- Social responsibility, impartiality
- Credibility, transparency, money traceability
- Safety, customer support, energy consumption
- Duration, eco footprint, yield, continuity
- Energy, availability, reliability, maintainability

Guess the domain!

Typical design aspects

- Decomposition
  - components, patterns
  - data/control flow, interfaces
- Software quality
  - deployment, life cycle, upgrades, lock in
  - resource usage, energy efficiency, time behaviour
  - error handling, data integrity, recovery

Typical realisation aspects

- Waterfall ⟺ Iterative
- Manual ⟺ Automated
- Buy ⟺ Reuse
- Adapt ⟺ Remake
- Feature first ⟺ Market first
- Develop ⟺ Outsource
- Assign ⟺ Hire
- Test for release ⟺ Test for integration

Specify domains, identify aspects

- What are your domains?
- What are aspects relevant for your project?
- Think of:
  - Application
  - Design
  - Realisation
- Just list them and give a short definition