PRAGMATIC ARCHITECTURE, TODAY

BART BLOOMMAERTS
@DAGGIEBe

SATURN, 2017-05-02
BART BLOOMMAERTS

- Application Architect
- Ordina Belgium
- JWorks
- @DaggieBe
WHAT? Architecture WHAT?
System

is made for

Stakeholder
VIEWPOINTS

- Context
- Functional
- Information
- Concurrency
- Development
- Deployment
- Operational
Architecture has an is documented in Architectural Description documents for Stakeholder uses Views are shaped by Perspectives
PERSPECTIVES

- Accessibility
- Availability and Resilience
- Development Resource
- **Evolution**
- Internationalization
- Location
- Performance and Scalability
- Regulation
- Security
- Usability
Architecture

REALLY?
ABILITY TO CHANGE
Architecture represents the significant design decisions that shape a system, where significant is measured by cost of change.

Grady Booch
A system is more than the sum of its parts; it is an indivisible whole. It loses its essential properties when it is taken apart.

Russell L. Ackoff
BE PART OF THE TEAM

Activity over role
AS A TEAM

- Facilitating
- Collaborating
- Modelling
- Sharing
- Resolving
- Improving
- ...
INVOLVED IN CODE?
YES

EXPERIENCE THE SAME PAIN AS EVERYBODY ELSE ON THE TEAM
PRAGMATIC ARCHITECTURE, TODAY

CODE

- Support and coaching
- Best practices
- Code reviews
- Proof of Concepts
- Reusable building blocks
- Etc.
SYSTEM
STAKEHOLDERS
EVOLUTIONARY
TEAM MEMBER
INVOLVED IN CODE
Architecture

VISUAL
One cannot see the ocean's currents by studying drops of water.

Grady Booch
VISUALISATION - GOAL

- Consistency
- Reporting
- Checking and validation
- **Sharing information**
VISUALISATION - GROUND RULES

- UML is usually not necessary
  - *Can* be used for model-driven development
- Boxes and lines
  - Be consistent
  - Provide a legend
  - Make sure stakeholders understand
VISUALISATION - GROUND RULES

- Decision log
  - Document your decisions and alternatives
- Avoid fluffy diagrams
  - Avoid mixed abstractions
- Only document what’s useful
WHY?

Architecture WHY?
WHY? UP-FRONT DESIGN
UP-FRONT DESIGN

- Think about the big picture
  - eg. used technology
  - automation
  - patterns (monolith, distributed, …)
  - layering
- And it’s evolution
- But … be pragmatic
Just enough up front design to create firm foundations for the software product and its delivery.

Simon Brown
JUST ENOUGH?
JUST ENOUGH?

- Greenfield
  - All the components of the system
  - High-level
    - Show the whole system
    - Add details later
  - White board diagram
JUST ENOUGH? GREENFIELD
JUST ENOUGH?

- Existing system
- As-is situation
- Domain of change
- Extension points
WHY? COMMUNICATION
COMMUNICATION

- During inception
  - Many stakeholders
  - Many desires
  - Many questions …
COMMUNICATION

- During development process
  - Sharing a technical vision
  - Everybody collaborates to the same end-goal
  - Leave your ego at the door
WHY? POLITICS
POLITICS

- Decisions
  - Stakeholders will question many decisions
  - Focus on system integrity
- Fear of change at corporate IT
  - Creating systems “nobody wants to touch”
- Play the game …
WHY? MAKE DECISIONS EXPLICIT
Architecture

HOW?
OBSERVE
DECIDE
ACT
OODA - EXAMPLE

- Context view
OODA - EXAMPLE

- Context view
OODA - EXAMPLE

- Context view

  - External System
  - System Frontend
  - System Backend
  - Database

  Arrows indicate data flow:
  - Send data
  - Request data
OODA - EXAMPLE

Context view

External System

External System Caller

System Frontend

System Backend

Database

request data

send data
RUNTIME OBSERVATIONS

- System composition
- Measuring value
- Application-driven
- Capturing data
READING MATERIAL

- Software Systems Architecture by Rozanski and Woods
- 37 Things One Architect Knows About IT Transformation by Gregor Hohpe
- Continuous architecture by Pureur and Erder
- Bredemeyer consulting - http://www.bredemeyer.com/
- Form follows function - https://genehughson.wordpress.com/
- Coding the architecture - http://www.codingthearchitecture.com/
- Continuous architecture in practice - https://pgppgp.wordpress.com/
THANK YOU
@
DAGGIEBE