



# A MEETING OF ZOMBIES

EINAR W. HØST

**ME**



# AGILE AND ARCHITECTURE



# STRIKING A BALANCE

# HOW MUCH ARCHITECTURE?

# HOW MUCH AGILITY?

# HOW MUCH AUTONOMY?



**« JUST ENOUGH »**



# THE GOLDILOCKS PRINCIPLE



**WHY?**

# WHY ARCHITECTURE?

# WHY AGILE?

**WHY DO ANYTHING?**

**WE SHOULD NEVER DO ANYTHING  
JUST BECAUSE WE ARE DOING IT**



**IF ARCHITECTURE IS THE ANSWER  
WHAT IS THE PROBLEM?**

**IF **AGILE** IS THE ANSWER  
WHAT IS THE PROBLEM?**

**IS IT THE SAME PROBLEM?**

# THE SOFTWARE CRISIS

**THE MAJOR CAUSE OF THE SOFTWARE CRISIS IS THAT THE MACHINES HAVE BECOME SEVERAL ORDERS OF MAGNITUDE MORE POWERFUL! TO PUT IT QUITE BLUNTLY: AS LONG AS THERE WERE NO MACHINES, PROGRAMMING WAS NO PROBLEM AT ALL; WHEN WE HAD A FEW WEAK COMPUTERS, PROGRAMMING BECAME A MILD PROBLEM, AND NOW WE HAVE GIGANTIC COMPUTERS, PROGRAMMING HAS BECOME AN EQUALLY GIGANTIC PROBLEM.**

THE MAJOR CAUSE OF THE SOFTWARE CRISIS IS THAT THE MACHINES HAVE BECOME SEVERAL ORDERS OF MAGNITUDE MORE POWERFUL BUT IT'S QUITE DIFFICULT AS LONG AS THERE WERE NO MACHINES, PROGRAMMING WAS NO PROBLEM AT ALL; WHEN WE HAD A FEW WEAK COMPUTERS, PROGRAMMING BECAME A MODERATE PROBLEM AND NOW WE HAVE GIGANTIC COMPUTERS, PROGRAMMING HAS BECOME AN EQUALLY GIGANTIC PROBLEM.

**WE CAN TRUST THE CRISIS  
TO REMAIN WITH US**

**AMBITIONS WILL CONTINUE TO RISE**



**EVER MORE UBIQUITOUS**

**EVER MORE SOPHISTICATED**

**MORE FOR LESS**

# TACKLING THE CRISIS

# STRUCTURE

# MODULAR DESIGN

# COMPONENTS

# REUSE



**THE 1990S WILL BE  
THE DECADE OF  
SOFTWARE ARCHITECTURE**



# RISE OF THE ARCHITECT

# WHAT IS ARCHITECTURE?

# IT'S WHAT ARCHITECTS DO

# WHAT DOES AN ARCHITECT DO?

**TENDS TO THE ARCHITECTURE!**

**THE ARCHITECTURE OF A SOFTWARE SYSTEM IS ITS ORGANIZATION OR STRUCTURE OF SIGNIFICANT COMPONENTS INTERACTING THROUGH INTERFACES, THOSE COMPONENTS BEING COMPOSED OF SUCCESSIVELY SMALLER COMPONENTS AND INTERFACES.**

THE ARCHITECTURE OF A SOFTWARE SYSTEM IS ITS ORGANIZATION OR STRUCTURE OF SIGNIFICANT COMPONENTS INTERACTING THROUGH INTERFACES, THOSE COMPONENTS BEING COMPOSED OF SUCCESSFULLY SMALL COMPONENTS AND INTERFACES.

# RUP



# RATIONAL UNIFIED PROCESS

# UNIFIED MODELLING LANGUAGE

# 4+1 VIEW MODEL

# PATTERNS MOVEMENT

# BIG BALL OF MUD

<http://www.laputan.org/mud/>

# ARCHITECTURE SCOPE CREEP

# ARCHITECTURE FOR SYSTEM PROPERTIES

# MORE ABSTRACT



**ARCHITECTURE REPRESENTS THE SIGNIFICANT DESIGN DECISIONS THAT SHAPE THE FORM AND FUNCTION OF A SYSTEM, WHERE SIGNIFICANT IS MEASURED BY THE COST OF CHANGE.**

# BOOOCH

**HOW DID IT GO?**



# AGILE THE ASSASSIN

**WHY?**

**UNCOMFORTABLE**

# IRRELEVANCE AND HARM

**MANY DEVELOPERS NEVER  
EXPERIENCED WORKING WITH  
AN ARCHITECT THAT DID  
MORE GOOD THAN HARM**



**WHY?**

# ARCHITECTURE SOUNDS GOOD

# WHY AN ARCHITECT?

# CONCEPTUAL INTEGRITY

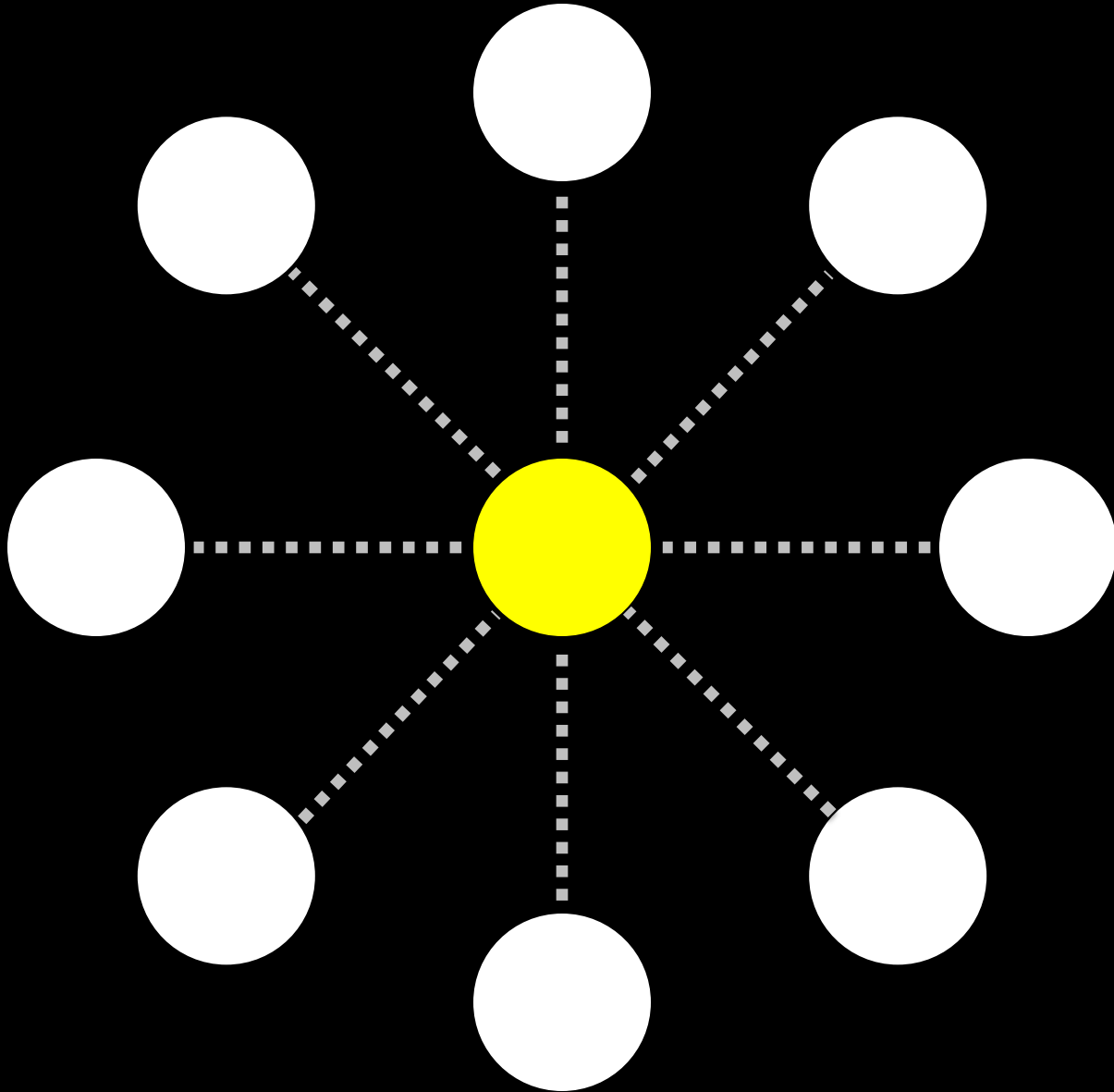
# MASTER BUILDER

# ECONOMY?

# SKILL COST

# ARCHITECT AS ANTI-PATTERN





# ARCHITECT AS BOTTLENECK

# ASYNCHRONOUS

**SLOW**

**DETACHED**

# POOR DECISIONS

# LOSS OF TRUST

# CONFLICTS OF INTEREST



# TEAMS

# **FIT-FOR-PURPOSE**

**SIMPLE**

**SPECIFIC**

**LOCAL**

**INTERNALLY LEGIBLE**

# ARCHITECT

# UNIFORM



# GENERAL

**EXTERNALLY LEGIBLE**

# SYSTEMIC PROBLEMS



# THE AGILE REVOLUTION

**INDIVIDUALS AND INTERACTIONS**  
**OVER PROCESSES AND TOOLS**  
**WORKING SOFTWARE**  
**OVER COMPREHENSIVE DOCUMENTATION**  
**CUSTOMER COLLABORATION**  
**OVER CONTRACT NEGOTIATION**  
**RESPONDING TO CHANGE**  
**OVER FOLLOWING A PLAN**

**INDIVIDUALS AND INTERACTIONS**

**OVER ~~PROCESSES AND TOOLS~~**

**WORKING SOFTWARE**

**OVER ~~COMPREHENSIVE DOCUMENTATION~~**

**CUSTOMER COLLABORATION**

**OVER ~~CONTRACT NEGOTIATION~~**

**RESPONDING TO CHANGE**

**OVER ~~FOLLOWING A PLAN~~**



# SCRUM + JIRA

# A PROCESS AND A TOOL?



# **INDIVIDUALS AND INTERACTIONS OVER PROCESSES AND TOOLS?**

**OH WELL**



# THE SCRUM PACT

# AGILE SCOPE CREEP

# AGILE SOFTWARE DEVELOPMENT

# AGILE PRODUCT DEVELOPMENT

# AGILE ORGANIZATIONS

# AGILE BUSINESS



**AN ENDLESS STREAM OF  
SILICON VALLEY SELF-HELP BOOKS  
PUSHING THE AGILE FRONTIER**



# RADICAL PIXIE DUST

**HAS IT WORKED?**

**WE'VE TRIED**

**EVERYTHING IS A SUCCESS  
UNTIL IT IS ABANDONED**

# VERY LITTLE STICKS

**THINGS HAVE GOTTEN WEIRD**

**THE IDEALS HAVE EVOLVED**



# BRIDGING THE GAP

# CUSTOMER OBSESSION

# SEAT AT THE TABLE

**REALITY IS LARGELY THE SAME**



# THE DEVELOPMENT TERRARIUM

# UNCOMFORTABLE TRUTHS



# COMMAND AND CONTROL NEVER DIED



**DON'T USE THAT FOREIGN WORD IDEALS**  
**WE HAVE THE EXCELLENT NATIVE WORD LIES**





**ARCHITECTURE IS DEAD**

# AGILE ISN'T AGILE

**NOW WHAT?**



# WHAT IS THE PROBLEM?



# THE PASSING OF TIME

**CHANGE**



# WHAT KINDS OF CHANGE?

# REQUIREMENTS

# DOMAIN CHANGES

# TECHNOLOGY

# CHANGE SURFACE

# ICEBERG APPLICATIONS

**PROBLEM**  
**CODE IS DEAD**

**CODE CANNOT RESPOND  
TO CHANGES BY ITSELF**



**CHANGES MUST BE  
RESPONDED TO BY PEOPLE**

**ONLY THE COMBINATION OF  
CODE AND PEOPLE IS ALIVE**



# SOFTWARE IS A CYBORG

# **SOCIO-TECHNICAL SYSTEM**

# **WHOLE NEW CATEGORIES OF CHANGE THAT AFFECT THE CYBORG**

**FUNDING IS CUT**

# TEAM CHANGES

# KNOWLEDGE LOSS



# SUSTAINABLE CYBORGS

**HUMAN VICIES ARE CONSTANT**

**SOLUTIONS MUST BE SYSTEMIC**

# ORGANIZATION LEVEL

# BEYOND THE TERRARIUM

# **TURN TERRARIUMS INTO ORGANIZATIONS**

# LARGER ORGANIZATION AS ALLIANCE OF SMALLER ORGANIZATIONS

**GIVE ACTUAL AUTONOMY**



**MOVE THE MONEY INSIDE**

**PUT NON-TECHIES INSIDE**

**PUT DECISIONS INSIDE**

# PUT OWNERSHIP INSIDE

# **DIRECT COMMUNICATION WITH USERS / CUSTOMERS**

**WITHOUT DIRECT COMMUNICATION  
FEEDBACK IS A BLUFF**

# TEAM LEVEL

**SEEK INDEPENDENCE**



# MINIMIZE DEPENDENCIES

# PROTECT INTERNALS

# RETAIN AND REBUILD KNOWLEDGE

# WORK IN GROUPS

# SEEK STABILITY

# **AVOID CHURN**

**REWRITE AS ONBOARDING**

# ARCHITECTURE LEVEL



# **FIT-FOR-PURPOSE**

# LOCAL LEGIBILITY

# OBSERVABILITY-DRIVEN

**YOUR SOFTWARE DOESN'T HAVE  
ANY QUALITY ATTRIBUTES THAT ARE  
NOT VALIDATED IN PRODUCTION**

# SIMPLICITY ( BUT FOR REAL )

# MINIMIZE MOVING PARTS

**GO BRUTALIST**

# EMBRACE CONSTRAINTS



# PRESERVE OPTIONALITY

# PLAN FOR DELETION

# BE VARY OF GENERALIZATIONS

# REDUCE CHANGE SURFACE

# SHRINK THE ICEBERG



# SUMMARY

**IT'S NOT A BALANCE**



# ARCHITECTURE BELONGS IN TEAMS

# SMASH THE TERRARIUM

# **BUILD SUSTAINABLE CYBORGS**