



# A MEETING OF ZOMBIES

EINAR W. HØST

**ME**



# Norkart

# AGILE **VS** ARCHITECTURE



# HOW MUCH ARCHITECTURE?

# HOW MUCH AGILITY?

# HOW MUCH AUTONOMY?



**“ JUST ENOUGH ”**





# 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  
QUITE UNFORTUNATELY AS THERE WERE NO  
MACHINES, PROGRAMMING WAS NO PROBLEM AT  
ALL; WHEN WE HAD A FEW WORK 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**

# HOW TO TACKLE 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.**

**RUUP**

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



# **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

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



BOOCH

# 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?

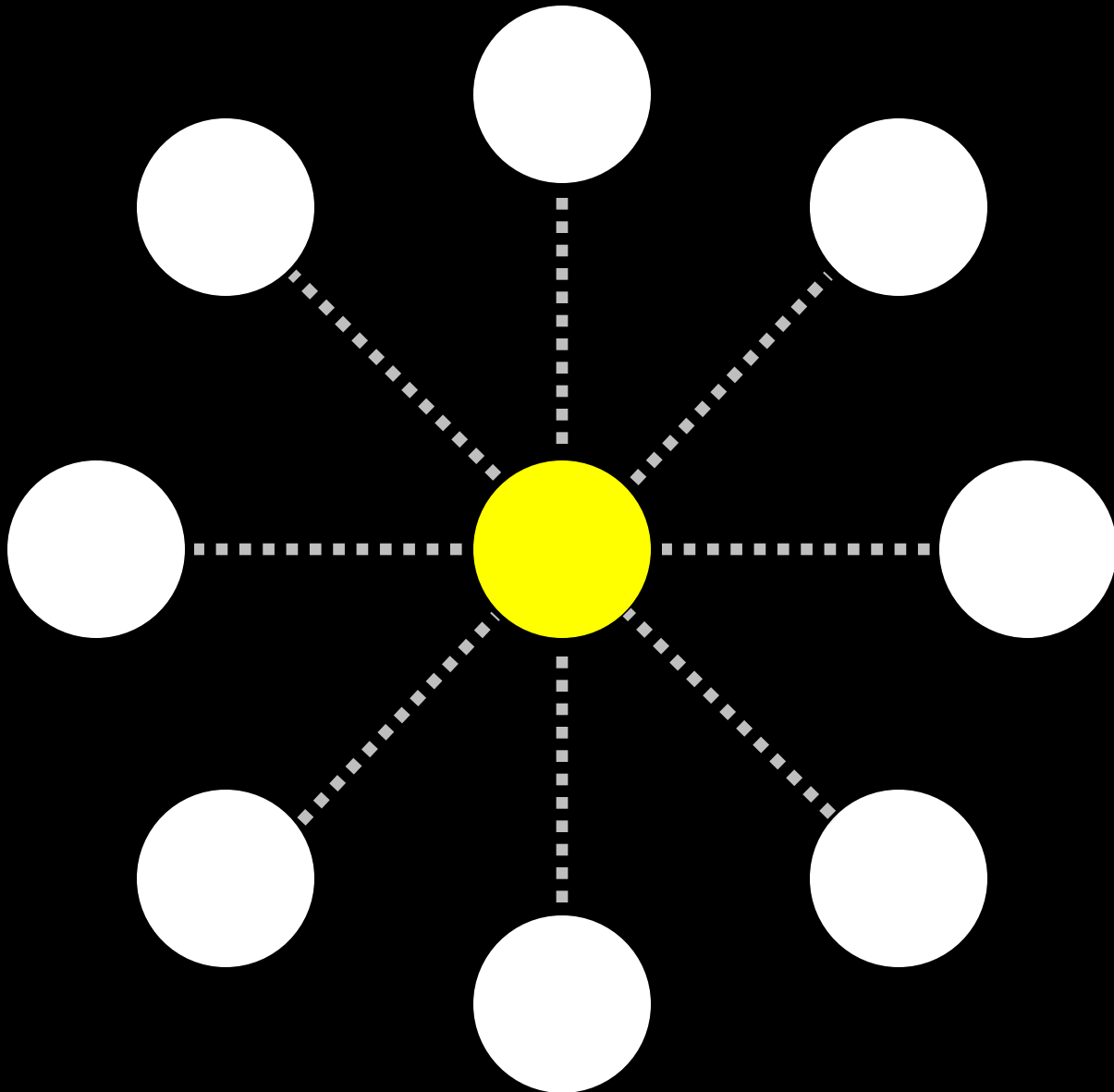
# CONCEPTUAL INTEGRITY

# MASTER BUILDER

# ECONOMY?

# SKILL COST

# ARCHITECT AS ANTI-PATTERN





# ARCHITECT AS BOTTLENECK

# ARCHITECT AS BOTTLENECK

ASYNCHRONOUS

# ARCHITECT AS BOTTLENECK

**SLOW**

# ARCHITECT AS BOTTLENECK

DETACHED

# ARCHITECT AS BOTTLENECK

**POOR DECISIONS**

# ARCHITECT AS BOTTLENECK

LOSS OF TRUST

# CONFLICTS OF INTEREST

# TEAM

# ARCHITECT



**TEAM**

**ARCHITECT**

**LOCAL**

**SPECIFIC**

**INTERNALLY LEGIBLE**

**TEAM**

**ARCHITECT**

**LOCAL**

**GLOBAL**

**SPECIFIC**

**GENERIC**

**INTERNALLY LEGIBLE**

**EXTERNALLY LEGIBLE**

**TEAM**

**ARCHITECT**

**LOCAL**

**GLOBAL**

**SPECIFIC**

**GENERIC**

**INTERNALLY LEGIBLE**

**EXTERNALLY LEGIBLE**

**NOT A RECIPE FOR SUCCESS!**

# SYSTEMIC PROBLEMS

A dramatic, painterly illustration of a revolutionary leader standing atop a crowd, holding a red flag, with a classical building in the background. The leader is a man with a determined, slightly bloodied face, wearing a dark, heavy coat. He stands with one arm raised, holding a large, tattered red flag that billows in the wind. Below him, a dense crowd of people, many wearing hats and coats, looks up at him with expressions of awe and anticipation. In the background, a grand, classical building with columns and arches is visible under a cloudy sky. The overall tone is one of intense historical drama and revolutionary fervor.

# 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~~**

**WORKING SOFTWARE**

**OVER ~~COMPREHENSIVE DOCUMENTATION~~**

**CUSTOMER COLLABORATION**

**OVER ~~CONTRACT NEGOTIATION~~**

**RESPONDING TO CHANGE**

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

**OH WELL**

# SCRUM



**still chaotic-good**

@TotherAlistair



Scrum struck a magnificent bargain in hostile territory:

Mgmt got 12 times/yr (1ce /mo) to change direction in any way they wanted;

Team got 1 mo of total quiet time w no interruptions or changes of direction to do heavy thinking/working.

No execs ever got a better deal.

7/

9:50 PM · Jun 16, 2021

A horror-themed illustration. In the center, a man with a pale, skull-like face and glowing red eyes sits at a desk, holding a pen over a document. To his right, a human skeleton sits at the same desk, looking towards the man. The scene is dimly lit, with a large, bright yellow full moon in the background. A black mug sits on the desk to the left. The overall style is dark and atmospheric, with a focus on the macabre.

# THE SCRUM PACT



# THE TERRARIUM MODEL OF SOFTWARE DEVELOPMENT





**WILL PRODUCE**  
**FEATURES**  
**~~SILK~~ FOR**  
**PROTECTION**

**WILL PRODUCE**  
**FEATURES**  
**~~SILK~~ FOR**  
**PROTECTION**

**WILL OFFER**  
**PROTECTION**  
**FEATURES\***  
**FOR ~~SILK~~**

**THAT WAS 20 YEARS AGO**

**AND THEN... NOTHING?**

**NOT QUITE**

# AGILE SCOPE CREEP

# AGILE SOFTWARE DEVELOPMENT

**AGILE SOFTWARE DEVELOPMENT**



**AGILE PRODUCT DEVELOPMENT**



**AGILE SOFTWARE DEVELOPMENT**



**AGILE PRODUCT DEVELOPMENT**



**AGILE ORGANIZATIONS**

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**

**THE IDEALS HAVE EVOLVED**

**“ BRIDGING THE GAP ”**

**THE IDEALS HAVE EVOLVED**

**“ CUSTOMER OBSESSION ”**

**THE IDEALS HAVE EVOLVED**

**“ SEAT AT THE TABLE ”**

**REALITY IS LARGELY THE SAME**



# THE DEVELOPMENT TERRARIUM





# COMMUNICATION THROUGH **MEDIUMS**



# UNCOMFORTABLE TRUTHS



**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

**THINGS HAPPEN**

**SOME OF THOSE THINGS  
AFFECT THE SYSTEMS WE MAKE**

**THE SYSTEMS MUST CHANGE  
TO REFLECT THE NEW REALITY**

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

**WE MUST MAKE IT**  
**APPEAR TO BE ALIVE**





CODE

PROGRAMMER





PROGRAMMER

CODE

**THIS IS THE SOFTWARE SYSTEM**



# SOFTWARE IS A CYBORG

**CHANGES MUST BE  
RESPONDED TO BY PEOPLE**

# CHANGE

**“ CHANGE IS CONSTANT ”**

# CHANGE

## WHAT ABOUT THE AMOUNT OF CHANGE?

# **CHANGE**

## **WHERE DO CHANGES COME FROM?**

# **CHANGE**

## **WHAT KINDS OF CHANGES?**

# **CHANGE**

## **HOW OFTEN?**



# **CHANGE**

## **SOURCES OF CHANGE**

# **CHANGE**

## **EXPOSURE TO CHANGE**

# CHANGE

## CHANGE SURFACE

# NECESSARY VS ACCIDENTAL

# **CAPACITY FOR CHANGE**

# CHANGES THAT AFFECT THE CODE

# CHANGES THAT AFFECT THE CODE

## REQUIREMENTS

# **CHANGES THAT AFFECT THE CODE**

## **BUSINESS DOMAIN**



# CHANGES THAT AFFECT THE CODE

## TECHNOLOGY

# CHANGES THAT AFFECT THE CODE

✨ TECHNOLOGY ✨

A conceptual image of an iceberg floating in dark blue water. The visible tip above the surface is small and jagged, while the submerged portion below the surface is much larger and more complex in shape, illustrating the concept of hidden potential or applications.

# ICEBERG APPLICATIONS

**EVERY DEPENDENCY IS  
A SUBSCRIPTION TO  
A STREAM OF CHANGES**

**ALL CHANGES TO DEPENDENCIES  
ARE ACCIDENTAL**

**WHAT'S THE PERCENTAGE OF TIME  
SPENT ON ESSENTIAL CHANGES?**

# **REDUCE EXPOSURE TO ACCIDENTAL CHANGES**

# **PROTECT OUR CAPACITY FOR CHANGE**



# SUSTAINABLE CYBORGS

**HUMAN VICES ARE CONSTANT**

**SOLUTIONS MUST BE SYSTEMIC**

# ORGANIZATION LEVEL

# BEYOND THE TERRARIUM

# BEYOND THE TERRARIUM

## SELF-SUSTAINING TEAMS

# BEYOND THE TERRARIUM

**TEAMS AS TINY ORGANIZATIONS**

# **BEYOND THE TERRARIUM**

**LARGER ORGANIZATION AS LOOSELY COUPLED ALLIANCE**



# ACTUAL AUTONOMY

# **ACTUAL AUTONOMY**

## **MOVE THE MONEY INSIDE**

# **ACTUAL AUTONOMY**

## **PUT NON-TECHIES INSIDE**

# **ACTUAL AUTONOMY**

## **PUT THE DECISIONS INSIDE**

# **DIRECT COMMUNICATION**

# **DIRECT COMMUNICATION**

## **WITH STAKEHOLDERS**

# **DIRECT COMMUNICATION**

## **WITH CUSTOMERS / USERS**

**WITHOUT DIRECT COMMUNICATION  
FEEDBACK IS A BLUFF**



# TEAM LEVEL

# INDEPENDENCE

# **INDEPENDENCE**

## **MINIMIZE DEPENDENCIES**

# **INDEPENDENCE**

## **WRITE SIMPLE THINGS YOURSELF**

# **INDEPENDENCE**

## **PROTECT INTERNALS**

# RETAIN AND REBUILD KNOWLEDGE

# **RETAIN AND REBUILD KNOWLEDGE**

**WORK IN GROUPS**

# ARCHITECTURE LEVEL



# TEAM ORIENTED

# **TEAM ORIENTED**

**ARCHITECTURE IS TOO IMPORTANT  
TO BE LEFT TO ARCHITECTS ALONE**

# TEAM ORIENTED

## LOCAL

**TEAM ORIENTED**

**SPECIFIC**

**TEAM ORIENTED**

**INTERNALLY LEGIBLE**

# CONSTRAINTS

# **CONSTRAINTS**

## **COMMUNICATION**

# CONSTRAINTS

## NEGOTIATION



# CONSTRAINTS

## REVIEW

# OBSERVABILITY-DRIVEN

# **OBSERVABILITY-DRIVEN**

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

**OBSERVABILITY-DRIVEN**

**BROADCAST THE STATE OF YOUR SYSTEM**

**OBSERVABILITY-DRIVEN**

**EXTERNALLY LEGIBLE**

# **SIMPLICITY**

## **(BUT FOR REAL)**

# **SIMPLICITY**

## **MINIMIZE MOVING PARTS**

# **SIMPLICITY**

## **AVOID SOPHISTICATION**



**SIMPLICITY**

**GO BRUTALIST**

# REDUCE CHANGE SURFACE

**REDUCE CHANGE SURFACE**

**SHRINK THE ICEBERG**



# SUMMARY

**IT'S NOT A BALANCE  
ARCHITECTURE BELONGS IN TEAMS  
SMASH THE TERRARIUM  
SHRINK THE ICEBERG  
BUILD SUSTAINABLE CYBORGS**

# SUMMARY

**IT'S NOT A BALANCE**

**ARCHITECTURE BELONGS IN TEAMS**

**SMASH THE TERRARIUM**

**SHRINK THE ICEBERG**

**BUILD SUSTAINABLE CYBORGS**

# SUMMARY

IT'S NOT A BALANCE

**ARCHITECTURE BELONGS IN TEAMS**

SMASH THE TERRARIUM

SHRINK THE ICEBERG

BUILD SUSTAINABLE CYBORGS

# SUMMARY

IT'S NOT A BALANCE  
ARCHITECTURE BELONGS IN TEAMS  
**SMASH THE TERRARIUM**  
SHRINK THE ICEBERG  
BUILD SUSTAINABLE CYBORGS



# SUMMARY

IT'S NOT A BALANCE  
ARCHITECTURE BELONGS IN TEAMS  
SMASH THE TERRARIUM  
**SHRINK THE ICEBERG**  
BUILD SUSTAINABLE CYBORGS

# SUMMARY

**IT'S NOT A BALANCE**  
**ARCHITECTURE BELONGS IN TEAMS**  
**SMASH THE TERRARIUM**  
**SHRINK THE ICEBERG**  
**BUILD SUSTAINABLE CYBORGS**

# SUMMARY

**IT'S NOT A BALANCE  
ARCHITECTURE BELONGS IN TEAMS  
SMASH THE TERRARIUM  
SHRINK THE ICEBERG  
BUILD SUSTAINABLE CYBORGS**