AN AGILE MDD APPROACH

A case study in a finance organization

Mina Boström Nakićenović
Sungard Front Arena, Stockholm

Introduction

- **Sungard**
  - large, world-wide financial services software company
  - 25 000 customers in 70 countries.
- **Front Arena system**
  - order management and deal capture on electronic exchanges
- **Market access**
  - based on a client/server architecture
- **Arena Market Servers – AMS**
  - provide services for market trading

Our Awards

2009 FINANCIAL NEWS EXCELLENCE IN I.T. TECHNOLOGY & TECHNOLOGY:
Best Systems Vendor.

2009 WATERS RANKINGS:
Best Counterparty Risk Solution Provider

2009 BANKING TECHNOLOGY "READERS' CHOICE" AWARDS:
Winner: Adaptiv: Best Risk Management Product (third year running).
Winner: Front Arena: Trade processing.
Runner up: Front Arena: Best Trading Platform / Order Management Product.

2009 CHARTIS RESEARCH RISKTECH100 RANKINGS:
SunGard placed first for Functionality and Trading and Capital Markets: ranked 3rd placed overall.

2008 CHARTIS RESEARCH RISKTECH100 RANKINGS:
SunGard ranked first place overall - 3rd consecutive year: Taking top honours in 5.

2009 FINTECH 100: SUNGARD TOP PROVIDER FOR CAPITAL MARKETS
(ranked 2rd in FinTech 100 top providers of IT solutions overall).

2008 RISK TECHNOLOGY RANKING (3RD CONSECUTIVE YEAR):
SunGard placed first for Functionality and Trading and Capital Markets: ranked 3rd placed overall.

2008 CREDIT MAGAZINE'S TECHNOLOGY INNOVATION AWARD:
SunGard placed first for Functionality and Trading and Capital Markets: ranked 3rd placed overall.

Mina Boström Nakićenović,
Sungard Front Arena

Sungard Front Arena

- One of the leading providers for the financial products and solutions
- Main requirements for staying on top of the competitive financial market:
  - Time-to-market
  - Quick responses on incoming changes (business and technical)

Agile Software Development

- It provides:
  - Qualitative and no cost-effective solutions
  - Quick delivery
- It welcomes:
  - Changing requirements even late in development
- It concerns different aspects:
  - Architectural
  - Organizational
  - Development process

Legacy Systems Evolution

- **Existing systems have to envolve constantly in order to continue to fulfill the incoming requirements/changes.**
- **How to use agile principles and techniques in a legacy system evolution?**

Case study

How Agile and Lean philosophy can help in producing an applicable solution?

Mina Boström Nakićenović,
Sungard Front Arena

Market Server Capability (MSC)

- **MSC is information about the trading functionallity that an electronic exchange (market) offers.**

- **MSC describes:**
  - **Market trading transactions (order, trade, ...)**
  - **Commands (enter, modify, delete,...)**
  - **Accessable attributes and fields (Price, Quantity, Trader,...)**

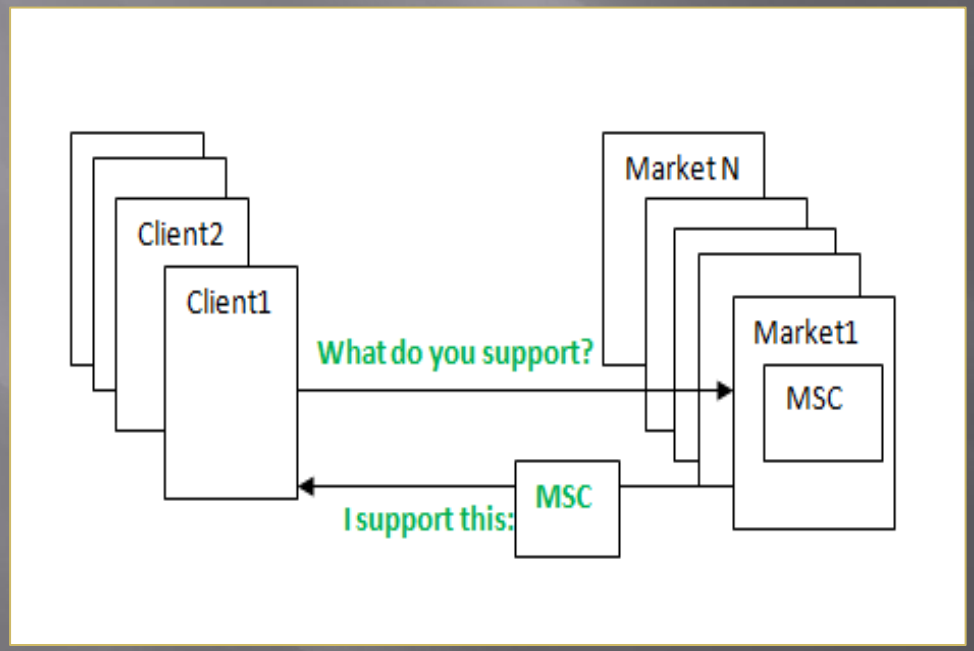
- **According to the MSC the access to the different market elements are permitted/disabled.**

The present MSC architecture

- Hard-coded MSC definition in each client
  - recompilation of components for each update
- Distributed definition
  - source files developed in the different programming languages (C++, C#, Java)
  - lack of cetralization
- Duplication of the MSC definition
  - the risk for data inconsistancy.
- No consistency control

Dynamic MSC

- Each AMS provides information about the MSC
- AMS sends dynamically MSC info to the clients.
- MSC definition is kept in the XML format



Transition Phase

- All AMS (20) and client components (6)
  - has to be upgraded
- The transition
  - occurs gradually
  - will take several years (different components' backlogs prioritizing).
- The hard-coded solution
  - has to be supported during the whole transition phase.
- An intermediate solution
  - should eliminate the duplication
  - should lead towards the DCM architecture

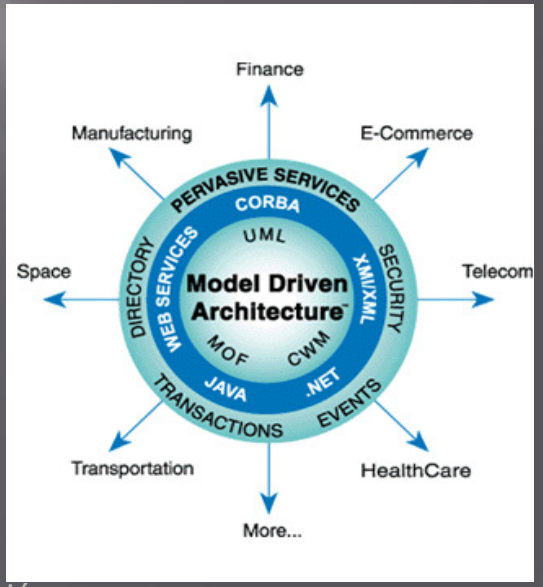
Intermediate Solution

3 points of wastes (Lean):

- Duplication
- Amount of work done
- Amount of time used for communication

MDA:

- Models
- Transformations
- Code generators
- Reverse engineering



Limitations

- Short time-frame
- No new tools or licenses
- No investment in change management

- Previous attempts of the MDD introduction failed
- UML modeling is not used in general

Agile MDD Approach

- "Think big, act small"
  - Long-term solution – DMC
  - Short-term solution – the intermediate solution
- Refactoring
  - Change the way how the developers work, not the way how the client components work
- "Simplicity is essential"
  - Simple is not simplistic!

Agile modeling and Code generators

Models

- MSC definition registry - XML format (good enough)
- Logical model – XSD schema
- PIM and PSM merged

Code generators

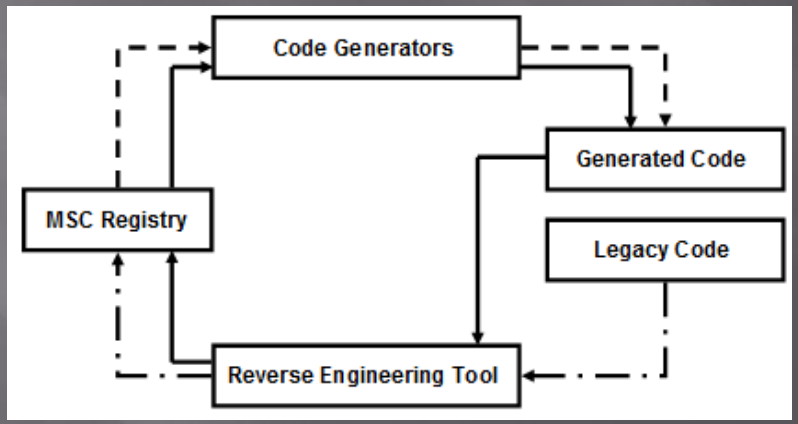
- XSL transformations
- ”Collective code ownership”

Reverse engineering of the Legacy code

- One-time process
  - Disabled on the end
- Our own tool
- ”Spike principle”
  - Start with simple scenarios
  - Do the whole round-tripping
  - Expand to the more complex scenarios in each sprint

Round tripping with the TDD approach

- Tests reflects parts of the ”Spike” (user stories)
- Reverse engineering and forward engineering are done together – one spike per sprint
- Round-tripping is performed at the end of each Sprint, using input from both the Legacy and the Generated code.



Benefits of TDD Approach

- **Development and testing in parallel**
  - **implementation phase was shortened**
- **No waste in the model**
  - **designed just according the output from the reverse engineering**
- **An efficient way to work**
- **Tests can be reused for the DMC testing**

Automation

On each MSC registry update:

1. **All MSC definition files are checked out**
2. **The trigger script invokes the code generators**
3. **All generated files are checked in**
4. **All affected components are recompiled**

Agile and Lean practices in MDD

Mina Boström Nakićenović,
Sungard Front Arena

Architectural aspects

- **"Eliminating waste"** (XP's Never duplicate your code)
  - One central model – the heart of the MDD
- **"Think big, act small"** (agile steps)
  - Intermediete solution can be improved to the long-term solution
- **"Simplicity is essential"**
  - The MDA concept reduced to the single modeling level

Organizational aspects

- ”Empower the team”
- ”Self-organizing teams”
- ”A constantly learning organization through relentless reflection and continuous improvement”

Development process aspects

- **"Spike principle"**
- **"Deliver as fast as possible"**
  - Agile principles made the starting curve for the MDD shorter.
- **"Wellcome changing requirements, even late in development"**
  - An iterative development, which allowed late model changes.
- **"Constant feedback"**
- **"Combine Scrum and Kanban process tools"**
- **"Reverse and forward engineering attain the same importance"**

Benefits of Agile MDD Approach

- Agile principles make the MDD starting curve shorter.
- Agile MDD approach instead of the full scale MDA.
- Agile modeling helps against building gargantuan models.
- Base your management decisions on a long-term philosophy even at the expence of short-term financial goals.
- TDD approach can contribute to the accelerated reengineering to the MDD solution.

Conclusion

Lean and Agile principles helped us with:

- **producing an intermediate solution**
- **coping successfully with the management constraints**
- **implementation within the short time-frame**
- **getting a legacy system more responsive to changes**
- **"Light" agile principles combined with a "heavy" MDA concept resulted in a pragmatic MDD solution**
- **This Agile MDD approach has been successfully applied on two projects by now.**

"It is not the strongest of the species that survive, nor the most intelligent, but the ones most responsive to change."

Charles Darwin, The Origin of the Species, 1859..