Lean Products

Rafael Sabbagh
Rodrigo de Toledo
Marcos Garrido
Carlos Felippe Cardoso
How successful are we?

a few numbers
Use of Software Features
Source: Standish Group, 2002 (?)
Software Pollution
Source: Cohen, Larson, Ware, 2001

- academic research
- 15 years
- 400 projects
- sizes from 10-500 person-years per release
- referenced by Jim Highsmith
Software Pollution
Source: Cohen, Larson, Ware, 2001

- Lines of code NOT deployed ("pollution")
- Lines of code deployed
Software Pollution
Source: Cohen, Larson, Ware, 2001

- 80% Features NEVER used
- 20% Features used
Success on Projects

Standish Group 2015: success means on time, on budget with a satisfactory result

- **Successful**: 29% (Agile: 39%)
- **Challenged**: 52% (Agile: 52%)
- **Failure**: 19% (Agile: 9%)
Can we become MORE successful?
A few simulation results
Pension Request Approval Process
Actual Project

- **stated goal**: automate pension approval process, which was entirely manual

- **actual project duration**: 2 years

- **obs**: originally planned for 6 months
Pension Request Approval Process
Simulation

- **actual value:** reduce errors

- **first release:** single calculation step responsible for ~50% of the errors

- **initial results:** 2 weeks to deliver 50% of the business value

- **later:** next steps causing more errors
Amazon International
Actual Project

**stated goal:** expand Amazon internationally, using local currency and tax system

**actual project duration:** 1 1/2 years
Amazon International Simulation

- **Actual value:** increase revenue through internationalization
- **First release:** start with China, believed to increase sales in 30%
- **Initial results:** 1 1/2 month to increase current value in 30%
- **Later:** prioritize region by market size?
Health Plan Provider Acquisition
Actual Project

- **stated goal:** integrate with acquired providers’ systems
- **actual project duration:** 4 years
- **obs:** four CTO's fired
Health Plan Provider Acquisition
Simulation

- **actual value:** reduce response time to approve procedures
- **first release:** shell to connect to systems via Selenium
- **initial results:** 1 1/2 month 70% of the business value
- **later:** transfer plans gradually to the acquirer original system
How can we become MORE successful?
Responsibility
NOT TICKET TAKERS
We MUST take the responsibility over defining the product!

It is OUR responsibility, NOT the customer’s.
Focus on Problems

Prioritize the Problems
Features are NOT problems!

Features are SOLUTIONS!
Identify problems to be solved, not features to be developed.

Seek for the underlying problem (root).

Suggestion: 5 why’s
We must understand what is the next most important problem to be solved.

Realize the vision incrementally, solving problem by problem.
Simplicity
Simplicity

From: Eric Burke, 2008
(originally http://stuffthathappens.com/blog/2008/03/05/simplicity)
Baby steps of value

“Simplest possible set of features that solve the problem”
Aim for the core of the core of the core of the core…
Produce and release in the shortest possible timeframe the simplest set of features that solve the next most important problem.

The core of the core of the core.

And again. And again.
Feedback
What happens when you put new features in front of the user?
FEEDBACK!

CHANGE!
Software development is an investigative work of hypothesis (in)validation!
Release very often and learn from the use!

Software development MUST be feedback-driven!

OBS: metrics are a very valuable form of feedback!
Focus on ROI
Return on the Investment
Return on the Investment

earliest point to stop the project?
Assess ROI continuously. Anticipate the end of the project to the earliest point where the vision is sufficiently realized.
Lean Products
a formal definition
Lean Product

“A product with no more and no less features than the necessary to achieve customer and user satisfaction through solving a problem or need.”

“Its process of production and delivery is such to avoid waste and maximize the return on the investments.”
Lean Product

Concepts built over Agile and Lean Startup principles and practices.

But taken to the extreme.
Lean Products Basic Principles

1. Start with what already works today, if anything.

2. Focus on problems to be solved, not on features to be developed.

3. The customer doesn’t know what she needs. Satisfy the customer by defining what will be developed next.

4. Create value in baby steps, using extreme simplicity in every increment of solution. Complexity comes incrementally, and only as needed.

5. Reduce risk by creating and taking every opportunity to deliver value. Develop each working solution in a very short time frame and put it to use.

6. Get as close as possible to the real use, measure it and understand it. The user can only provide true feedback after using it.
A few more simulation results
Driver’s License Exam
Actual Project

- **stated goal**: enable candidates to do theoretical exam online
- **actual project duration**: 6 months
- **obs**: outsourced company gets bonus by # of exams applied
Driver’s License Exam Simulation

- **actual value:** allow instant results
- **first release:** static html exam for instant results. Inspector gathers scores manually
- **initial results:** 1 week to deliver 40% of business value
- **later:** more static html exams, then randomize questions from DB, then forms for new questions etc.
Access to Legal Process Documents

Actual Project

**Stated goal:** enable law firms to integrate their systems to get legal process data

**Actual project duration:** 5 years

**Obs:** huge government project
Access to Legal Process Documents
Simulation

- **actual value:** drastically reduce time to access the documents
- **first release:** scan, OCR and index all files, available as webservice for search
- **initial results:** 4 weeks to deliver 80% of the business value
- **later:** alerts on changes, history, document upload
Access to Legal Process Documents

Actual Project

- **stated goal:** enable law firms to integrate their systems to get legal process data
- **actual project duration:** 5 years
- **obs:** huge government project
Access to Legal Process Documents

Simulation

- **Actual value:** drastically reduce time to access the documents

- **First release:** scan, OCR and index all files, available as webservice for search

- **Initial results:** 4 weeks to deliver 80% of the business value

- **Later:** alerts on changes, history, document upload
Job Placement for Strategic Positions

Actual Project

illezd goal: provide set of recommended people for open seats

actual project duration: 1 year

obs: existing CV project
Job Placement for Strategic Positions
Simulation

- **actual value**: ease and better target decision for open seats

- **first release**: use existing CV system; append needed filterable fields; force everyone to update

- **results**: 1 month and project done
Identifiable Patterns
an incomplete list
Pattern: End to End Dimmer

- **Description**: flow with multiple steps or actions, currently either software, manual or inexistent. Aims to solve an end to end problem. Most usual pattern

- **Proposed solution**: solve the end to end problem from the beginning with the minimum necessary and sufficient (critical steps). Improve problem resolution with dimmer

- **Example**: any online store

- **Simulation**: Driver’s License Exam and Access to Legal Process Documents
Pattern: Metric Improvement Dimmer

- **Description:** existing flow with multiple steps or actions, currently software or manual. Aims to improve a metric (e.g. errors, sells, risk on code maintainability)

- **Proposed solution:** dimmer; replace flow step/action by flow step/action with new software; prioritize from the step or action which better improves the metric

- **Example:** automate a manual process
- **Example:** reformulated version of a working software

- **Simulation:** Pension Request Approval Process
Pattern: 1, 2, n

- **Description**: system to work on multiple areas with similar *modus operandi*, but different rules. Currently, software or inexistent. Aims to improve (expand) a metric

- **Proposed solution**: Prioritize based on improving the metric. Choose 1st area. Create specific solution. Choose 2nd. New specific solution. Choose next. Now, create generic solution, using learning from 1, 2 and 3 as guide

- **Example**: whole module used for a specific purpose, now to be used in different places

- **Simulation**: Amazon International
Pattern: Process Extension

• **Description:** working process which needs to be extended, currently software or manual

• **Proposed solution:** append to what currently works just the sufficient to solve the problem (e.g. new database fields, filters, screens etc.)

• **Example:** old software, small changes will answer to broader needs

• **Simulation:** Job Placement for Strategic Positions
Pattern: External Shell

- **Description:** multiple data sources to be unified/integrated and accessed by a common interface

- **Proposed solution:** create a shell to access the multiple data sources from a single interface. If necessary, use a migration strategy for the data (e.g. little by little to one of the sources), gradually removing the other data sources

- **Example:** merger or acquisition with different running systems with data that needs to be integrated

- **Simulation:** Health Plan Provider Acquisition
Thank you!

Rafael Sabbagh
rsabbagh@k21.com.br
linkedin.com/in/rafaelsabbagh

Rodrigo de Toledo
rtoledo@k21.com.br
linkedin.com/in/rodrigodetoledo