MADS

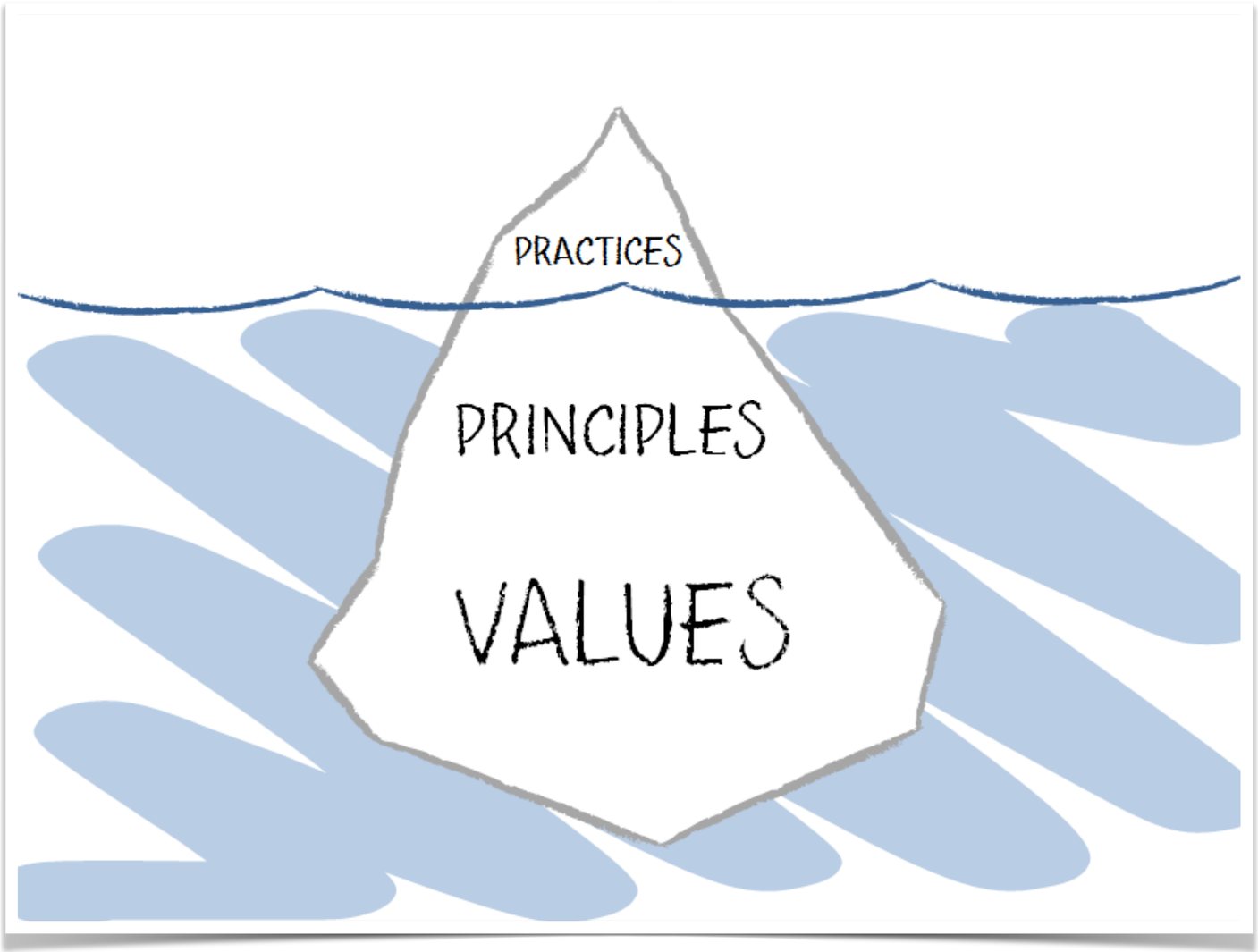
S4: El Manifiesto Agil

Índice

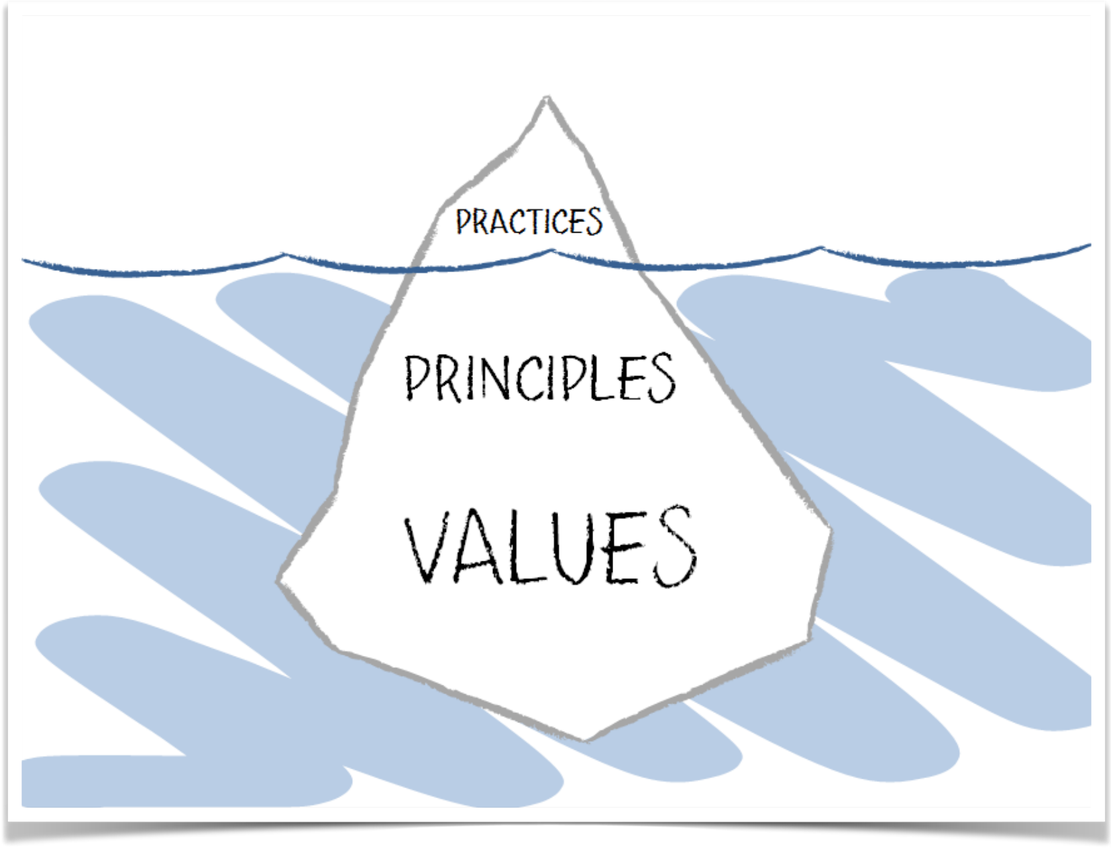
Valores, principios y prácticas: Ejemplo centro de secundaria

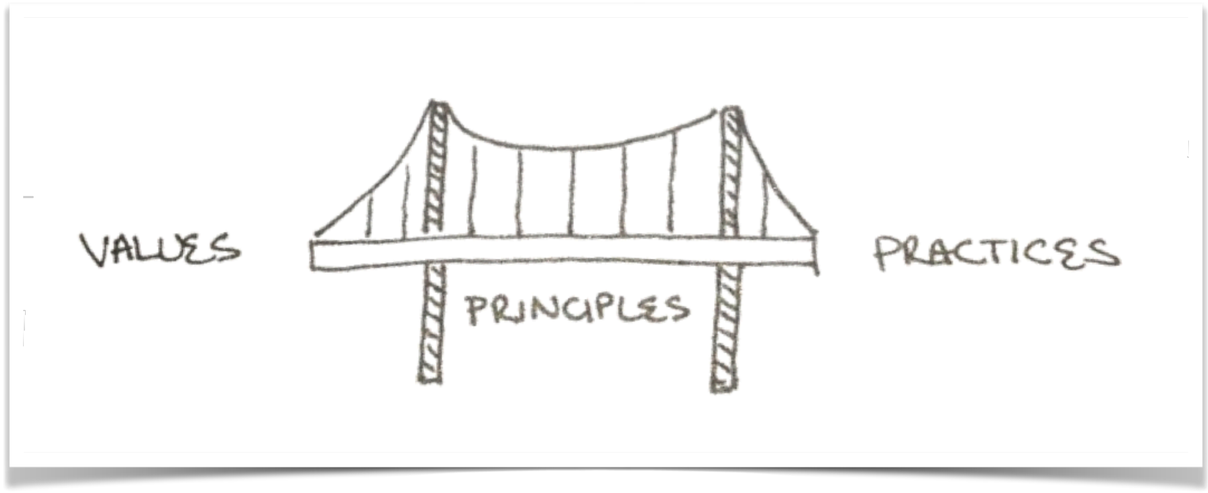
- **Valores**
  - “Nuestros estudiantes se comprometen con su trabajo”
  - “Creemos en la responsabilidad individual más que en la obediencia a un proceso”
- **Principios**
  - “Entregar los trabajos a tiempo”
  - “Los trabajos deben ser originales”
  - “La asistencia a clase es obligatoria”
- **Prácticas**
  - “Encargamos un trabajo individual sobre la biografía de un/a científico/a que debe ser entregado en 20 días”
  - “Exposición en clase de los trabajos. Se votan los mejores y se entregan unos premios (libros)”

Valores, principios y prácticas

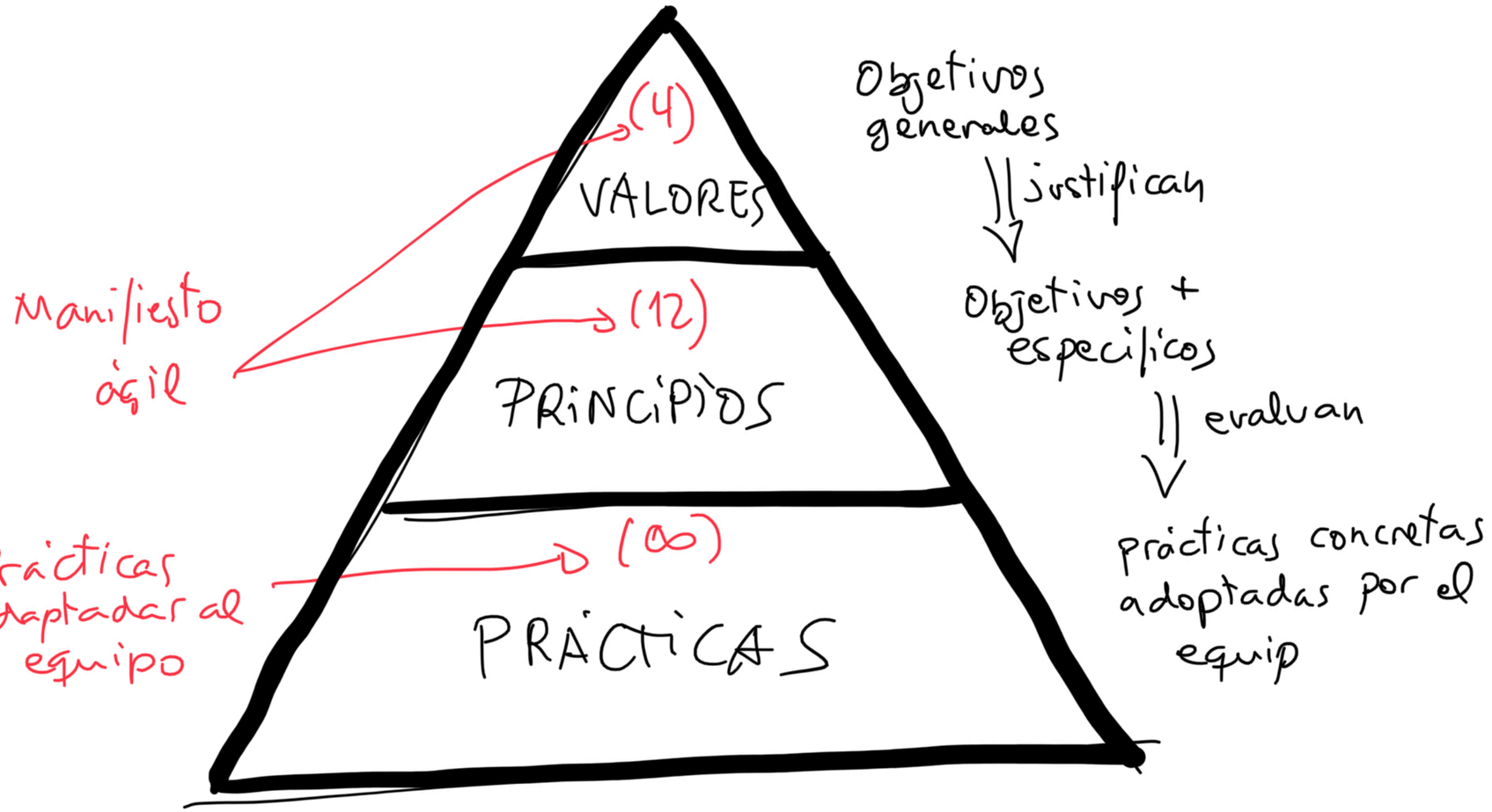


Valores, principios y prácticas



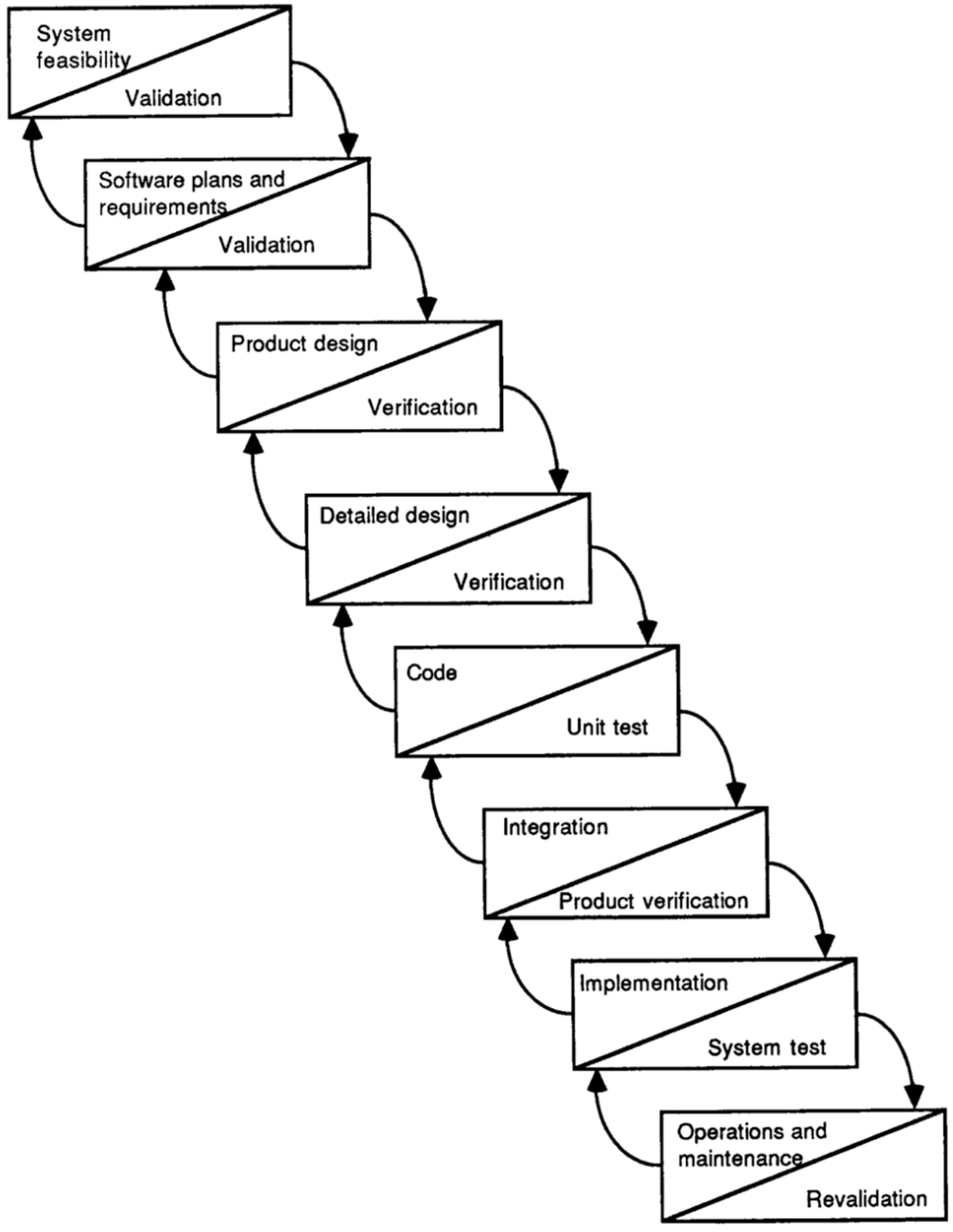


Valores, principios y prácticas

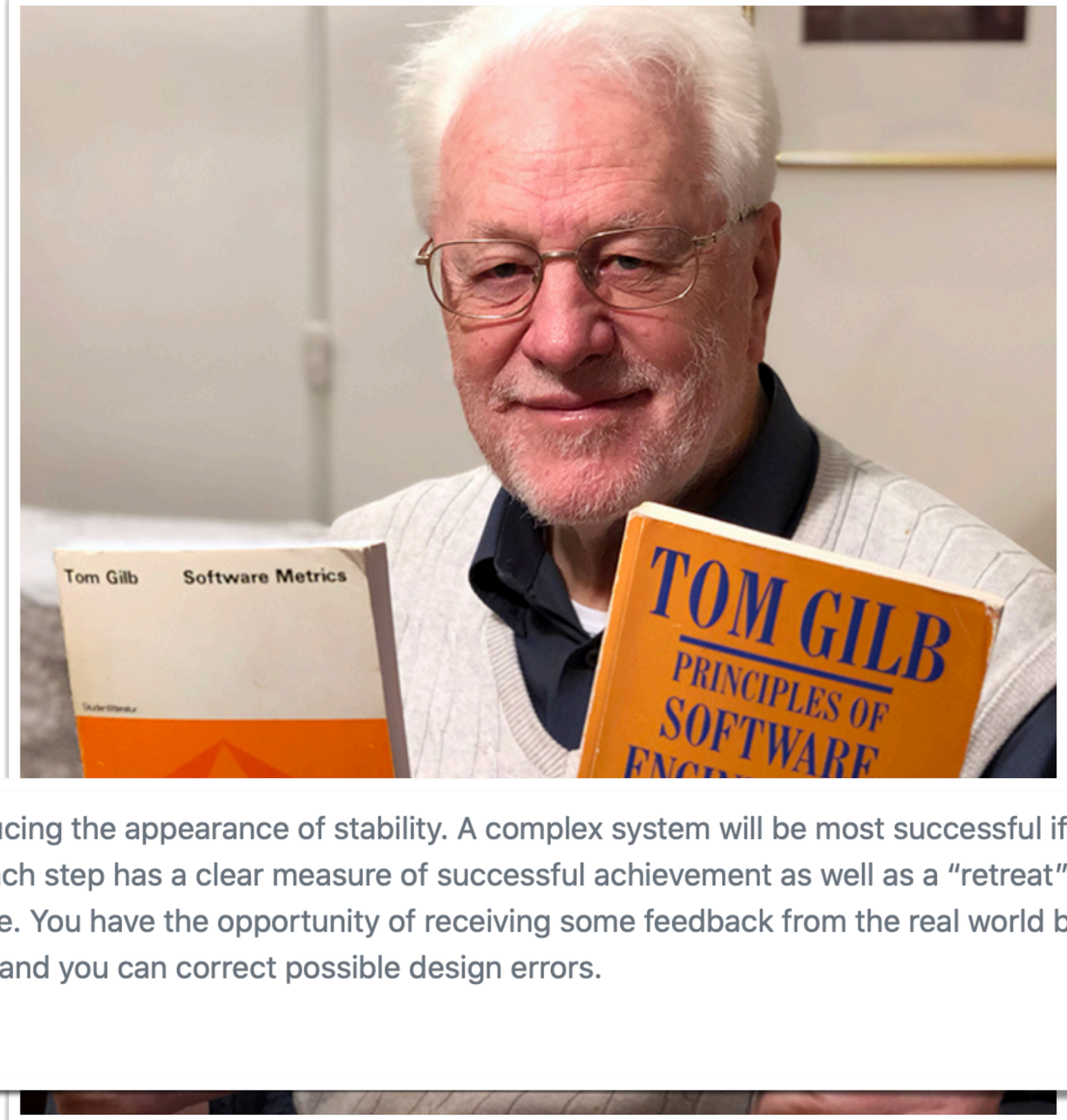


2. Tendencias previas al manifiesto ágil

1970


System feasibility
Validation
Software plans and requirements
Validation
Product design
Verification
Detailed design
Verification
Code
Unit test
Integration
Product verification
Implementation
System test
Operations and maintenance
Revalidation

1976



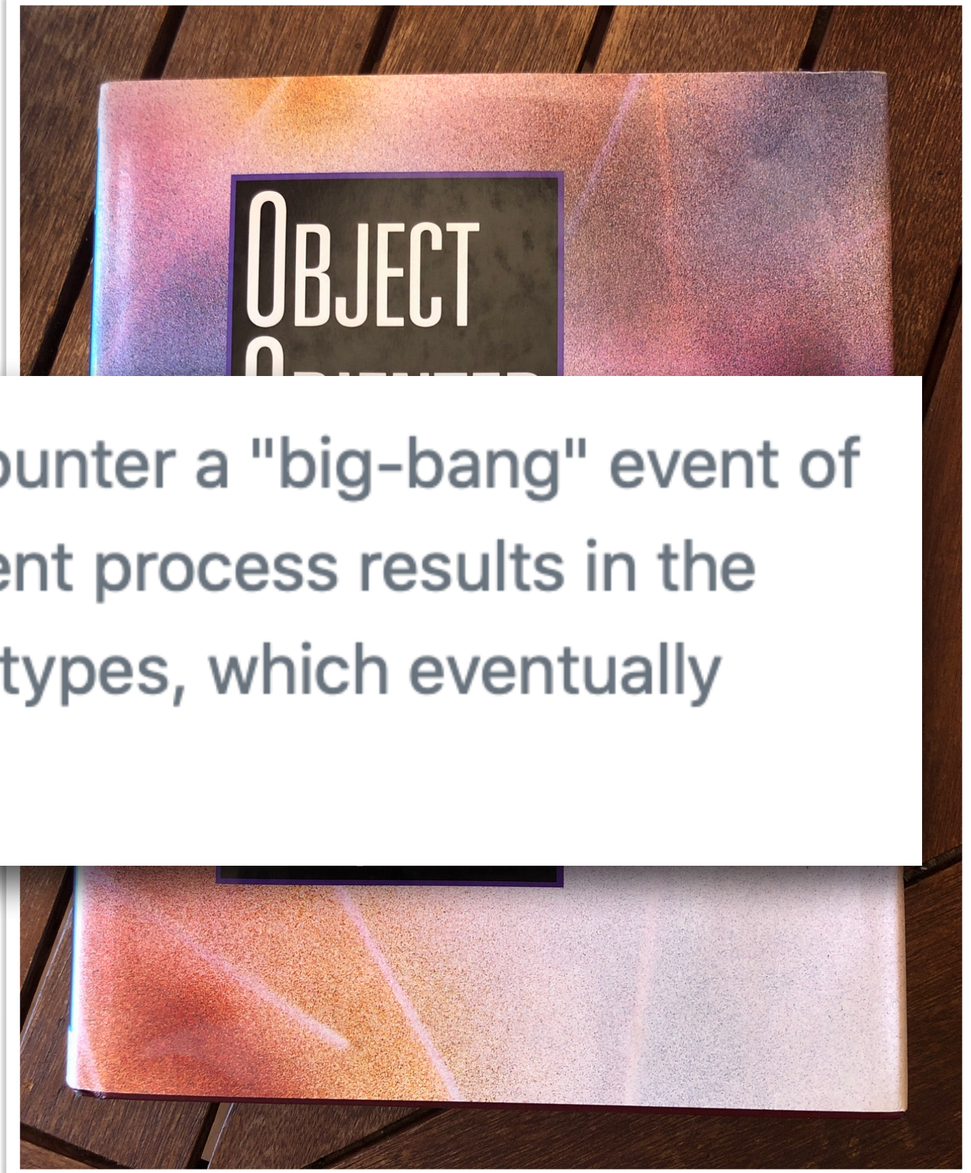
""Evolution" is a technique for producing the appearance of stability. A complex system will be most successful if it is implemented in small steps and if each step has a clear measure of successful achievement as well as a "retreat" possibility to a previous successful step upon failure. You have the opportunity of receiving some feedback from the real world before throwing in all resources intended for a system, and you can correct possible design errors.

Tom Gilb (Software Metrics, 1976)

1980

```
DISPLAY  CUSTOMER  INFORMATION                    Acct = B10093

  Credit Limit:$     0     Finance Charge? Y    Area:        Sort Codes: B

                    BILLING                              SHIPPING
    Name: A CLEAN WELL LIGHTED PLACE FOR     Name: A CLEAN WELL LIGHTED PLACE FOR
 Address: 601 VAN NESS AVENUE             Address: 601 VAN NESS AVENUE
        :                                        :
        :                                        :
    City: SAN FRANCISCO                      City: SAN FRANCISCO
   State: CA                                State: CA
     Zip: 94102                               Zip: 94102
 Country: U.S.A                           Country:

   Phone:                                   Phone:




          Enter ↑ to skip back, ↓ to skip forward, or <ESC> to exit
```

1980

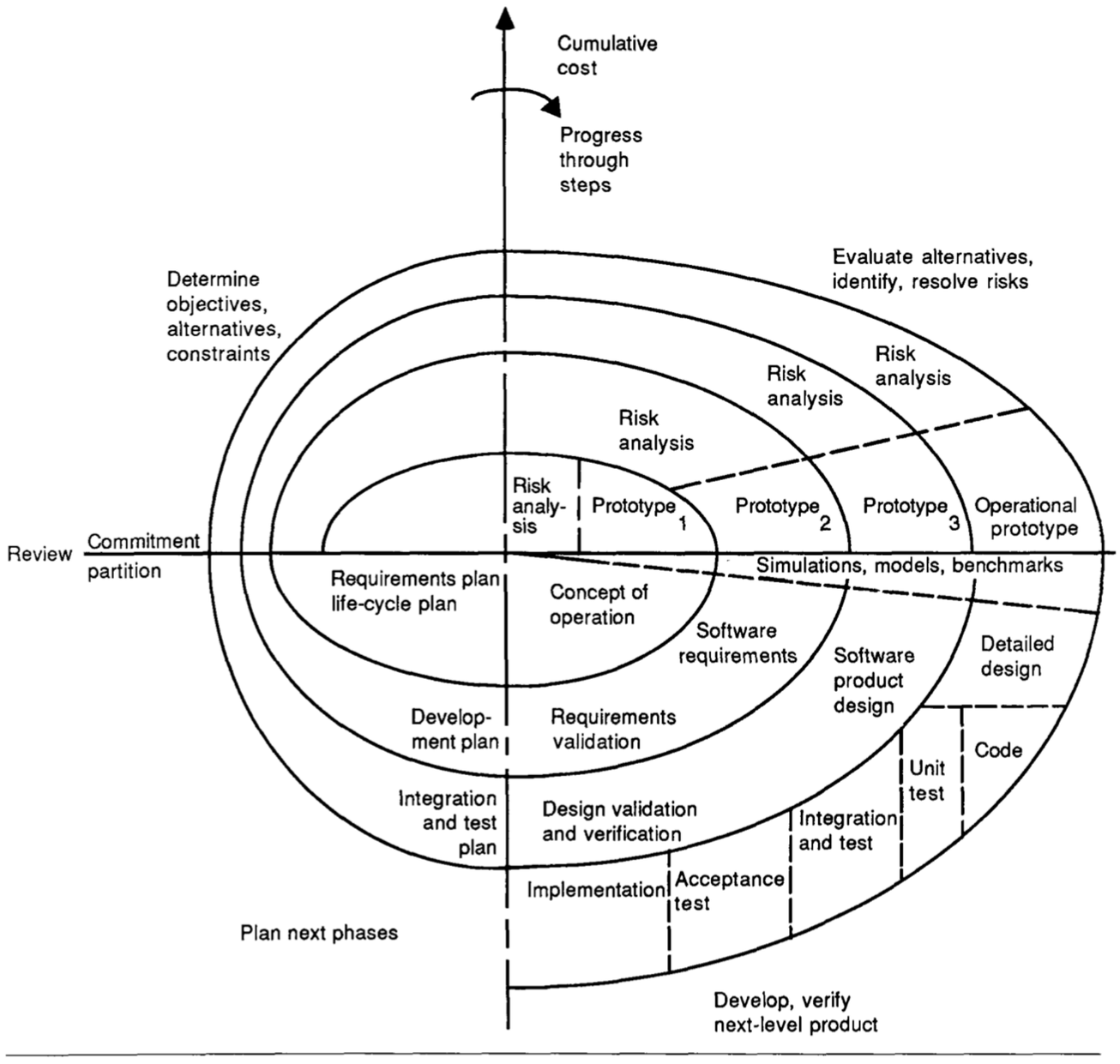


With object-oriented design we never encounter a "big-bang" event of system integration. Instead, the development process results in the incremental production of a series of prototypes, which eventually evolve into the final implementation.

Object Pascal

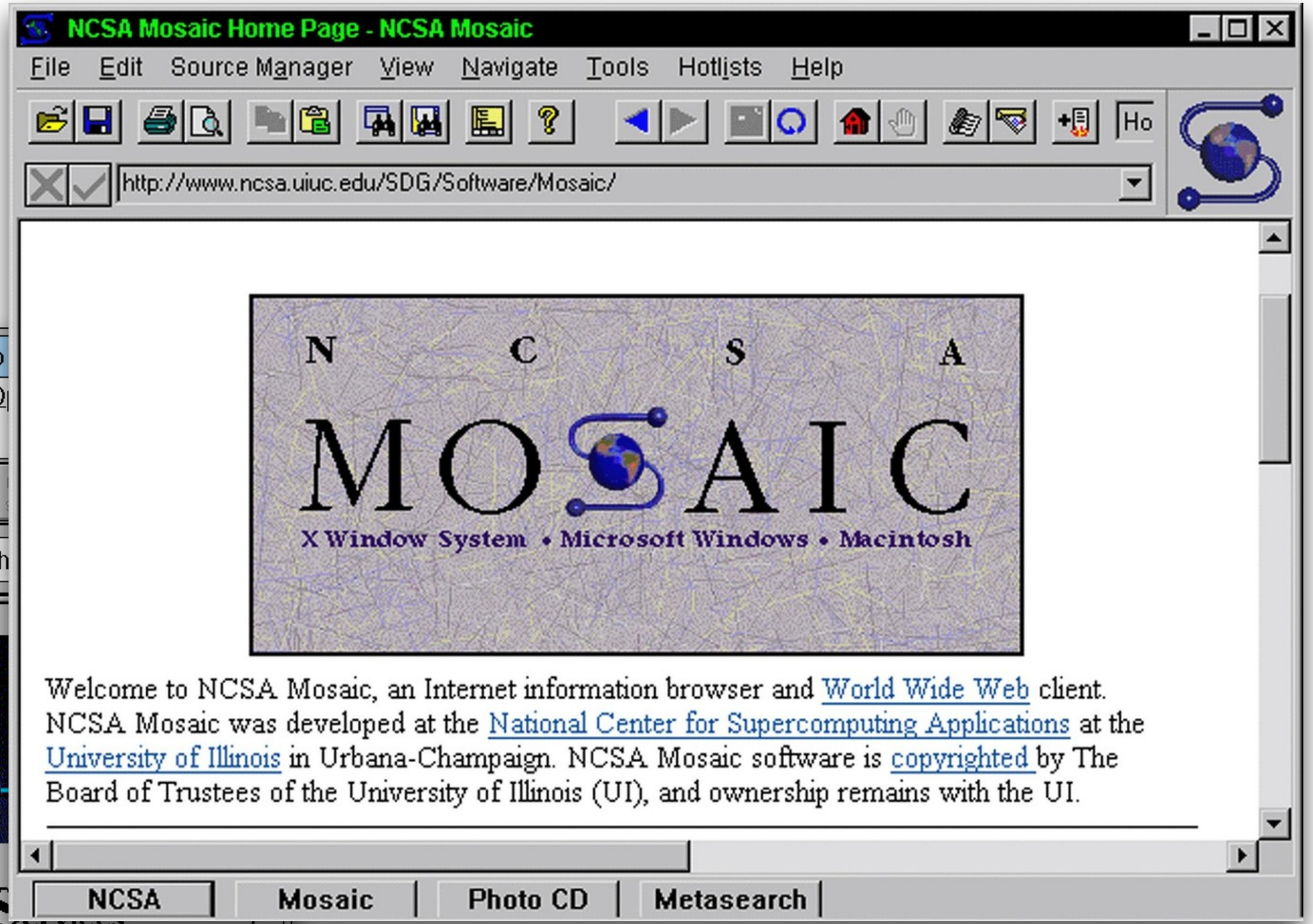
1985

Bohem Spiral model

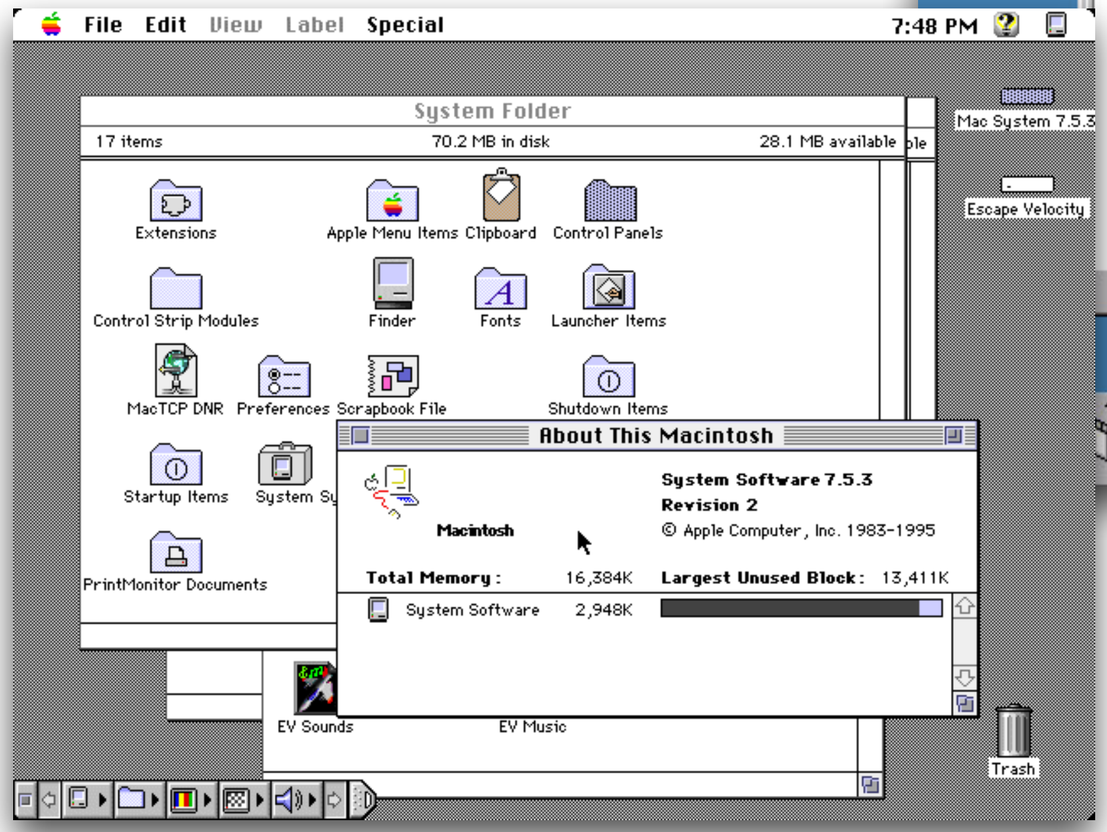


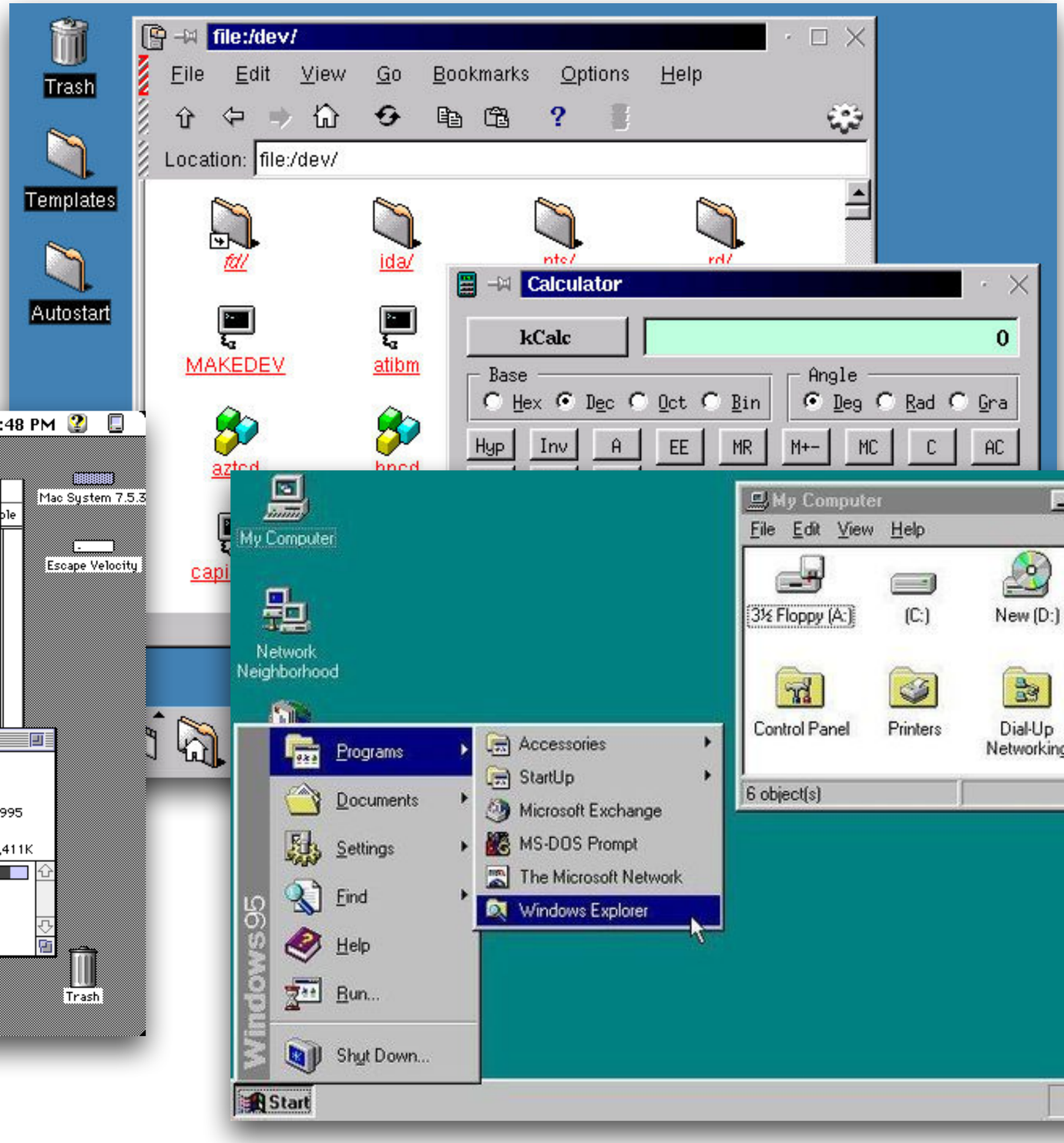
1993


How to Create World Wide Web
File Edit View Go Bookmarks
Back Forward Home Reload Images Open Find
Location: http://home.mcom.com/home/h
WELCOME TO
How to Create WWW S
So you've been wandering the web for a while now, and you're ready to start contributing to the great flow of information on the Internet. Where do you start? How do you begin? Providing information on the World Wide Web has two parts: writing the World Wide Web


NCSA Mosaic Home Page - NCSA Mosaic
File Edit Source Manager View Navigate Tools Hotlists Help
http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/
N C S A
MOSAIC
X Window System • Microsoft Windows • Macintosh
Welcome to NCSA Mosaic, an Internet information browser and World Wide Web client. NCSA Mosaic was developed at the National Center for Supercomputing Applications at the University of Illinois in Urbana-Champaign. NCSA Mosaic software is copyrighted by The Board of Trustees of the University of Illinois (UI), and ownership remains with the UI.
NCSA
Mosaic
Photo CD
Metasearch

1995

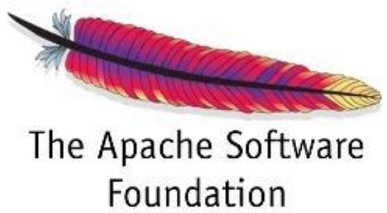

File Edit View Label Special
7:48 PM
System Folder
17 items
70.2 MB in disk
28.1 MB available
Mac System 7.5.3
Escape Velocity
Extensions
Apple Menu Items
Clipboard
Control Panels
Control Strip Modules
Finder
Fonts
Launcher Items
MacTCP DNR
Preferences
Scrapbook File
Shutdown Items
Startup Items
PrintMonitor Documents
About This Macintosh
System Software 7.5.3
Revision 2
Macintosh
© Apple Computer, Inc. 1983-1995
Total Memory: 16,384K
Largest Unused Block: 13,411K
System Software 2,948K
EV Sounds
EV Music
Trash


Trash
Templates
Autostart
file:/dev/
File Edit View Go Bookmarks Options Help
Location: file:/dev/
fd//
ida/
MAKEDEV
atibm
Calculator
kCalc
0
Base
Hex Dec Oct Bin
Angle
Deg Rad Gra
Hyp Inv A EE MR M+- MC C AC
My Computer
Network Neighborhood
Programs
Documents
Settings
Find
Help
Run...
Shut Down...
Windows95
Accessories
StartUp
Microsoft Exchange
MS-DOS Prompt
The Microsoft Network
Windows Explorer
My Computer
File Edit View Help
3½ Floppy (A:)
(C:)
New (D:)
Control Panel
Printers
Dial-Up Networking
6 object(s)
Start
1:22 PM

1990s


MySQL


odoo


The Apache Software
Foundation

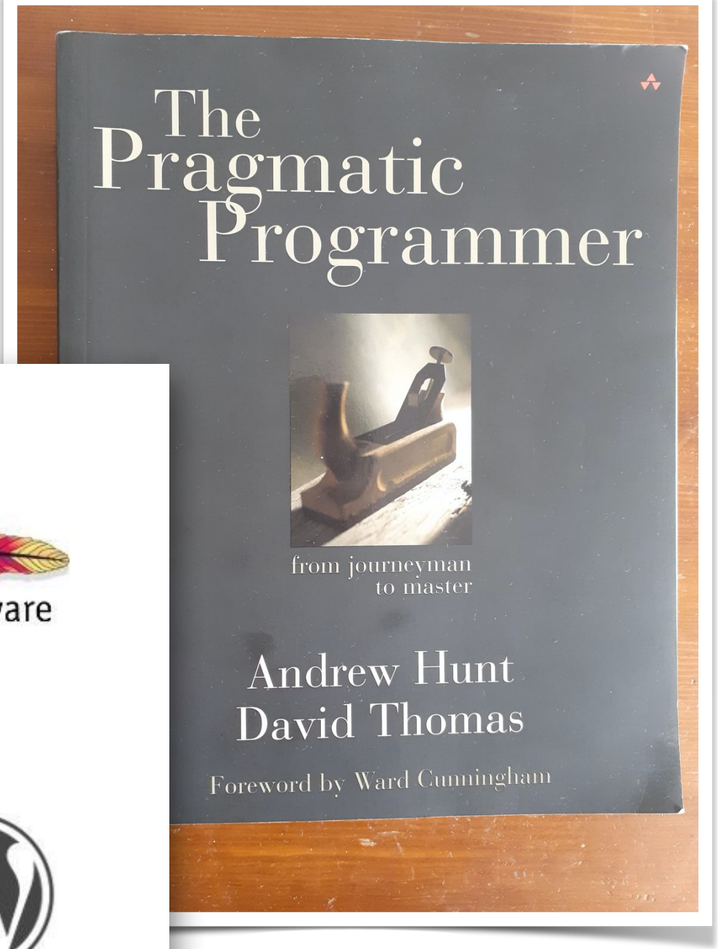

Joomla!™
...because open source matters


Apache
OpenOffice™


open source
initiative


Magento™
Open Source eCommerce


Java


The
Pragmatic
Programmer
from journeyman
to master
Andrew Hunt
David Thomas
Foreword by Ward Cunningham

1990s

Iterative and Incremental Development: A Brief History

Although many view iterative and incremental development as a modern practice, its application dates as far back as the mid-1950s. Prominent software-engineering thought leaders from each succeeding decade supported IID practices, and many large projects used them successfully.

Craig Larman
Valtech

Victor R. Basili
University of Maryland

As agile methods become more popular, some view iterative, evolutionary, and incremental software development—a cornerstone of these methods—as the "modern" replacement of the waterfall model, but its practiced and published roots go back decades. Of course, many software-engineering students are aware of this, yet surprisingly, some commercial and government organizations still are not.

PRE-1970

IID grew from the 1930s work of Walter Shewhart,[1] a quality expert at Bell Labs who proposed a series of short "plan-do-study-act" (PDSA)

https://doi.org/10.1186/s40411-018-0059-z *ering Research and Development* (2018) 6:15
https://doi.org/10.1186/s40411-018-0059-z

Journal of Software Engineering Research and Development

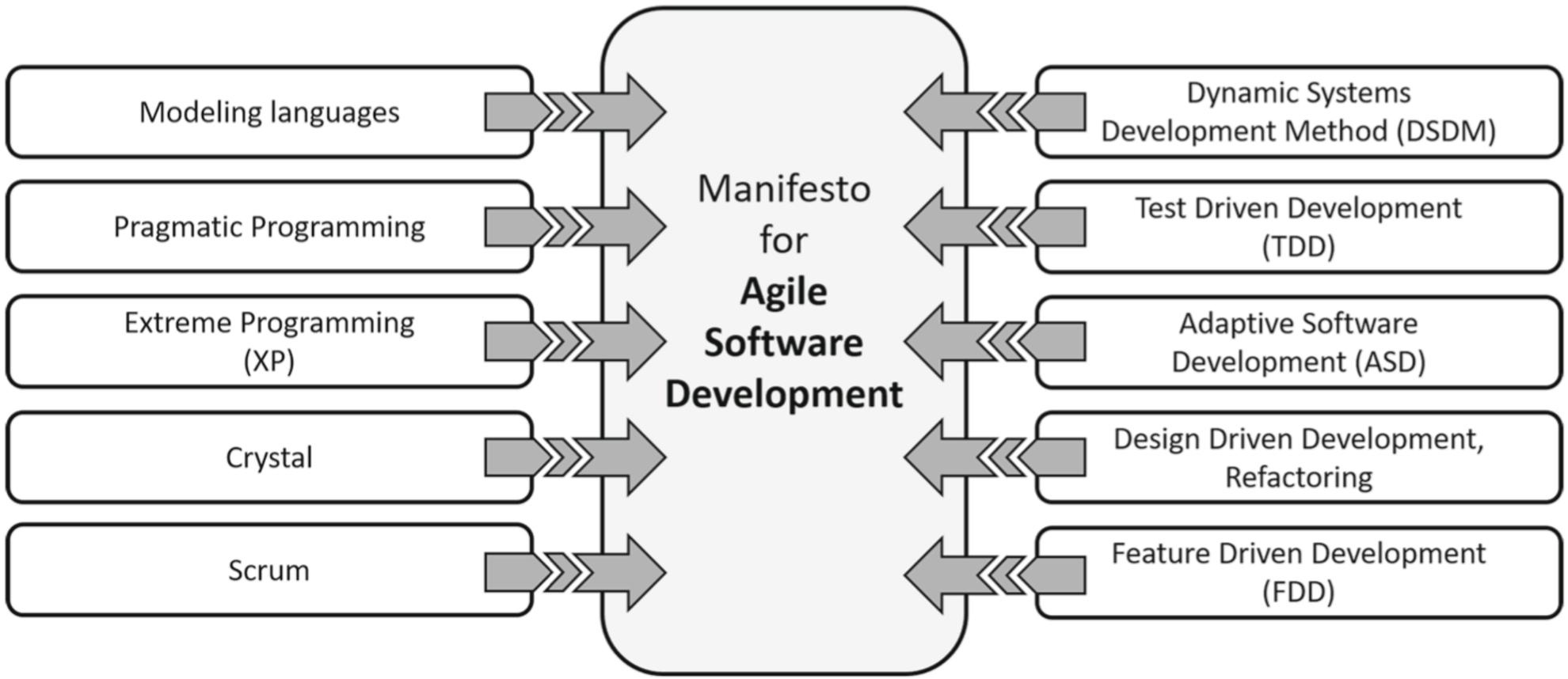
RESEARCH Open Access

Back to the future: origins and directions of the "Agile Manifesto" – views of the originators

Philipp Hohl[1*], Jil Klünder[2], Arie van Bennekum[3], Ryan Lockard[4], James Gifford[5], Jürgen Münch[6], Michael Stupperich[1] and Kurt Schneider[2]

*Correspondence:
philipp.hohl@daimler.com
[1]Daimler AG,
Research and Development,
Wilhelm-Runge-Str. 11, 89081 Ulm,
Germany
Full list of author information is available at the end of the article

Abstract
In 2001, seventeen professionals set up the manifesto for agile software development. They wanted to define values and basic principles for better software development. On top of being brought into focus, the manifesto has been widely adopted by developers, in software-developing organizations and outside the world of IT. Agile principles and their implementation in practice have paved the way for radical new and innovative ways of software and product development. In parallel, the understanding of the manifesto's underlying principles evolved over time. This, in turn, may affect current and



2000

Writing The Agile Manifesto

Origins

I can fairly accurately ascribe the origins of the Agile Alliance get together to a retreat held for various leaders in the Extreme Programming community in the Spring of 2000. Kent invited a bunch of active XPers to his rural part of Oregon to discuss various issues in XP. As well as confirmed XPers he also invited a number of people who were interested but separate to XP: such as Alistair Cockburn, Jim Highsmith, and Dave Thomas.

At the meeting we discussed the relationship between XP and other methods of a similar ilk - at the time referred to as Lightweight Methods. We agreed that XP was best as a specific process: "a stake in the ground". We also agreed there was a lot of common ground between XP and some of these other methods. As a result of this (Uncle) Bob Martin decided to try to put together a meeting of people interested in this broader range of methods.

2001

November 20, 2000

snowbird
ski and summer resort

Allister Cockburn
Humans and Technology
7691 Dell Road
Salt Lake City, UT 84121
Telephone: 801-947-9275
Fax: 775-416-6457

Snowbird Sales Contact:
Jim Dixon
Executive Meeting Manager
Telephone: 801-933-2272
Fax: 801-933-2298

RE: Light Weight Methods Conference
CONTRACT DUE DATE: November 27, 2000

The following arrangements and conditions will become binding between Snowbird Corporation and Humans and Technology upon signature and receipt to Snowbird by **November 27, 2000.**

DATES:	**Arrival:**	Sunday, February 11, 2001
	Departure:	Wednesday, February 14, 2001

3. El Manifiesto Agil

17 Firmantes

XP

- Kent Beck (*Twitter*)
- Robert C. Martin (*Twitter*, *Clean Code*)
- Ward Cunningham (*Twitter*)
- Ron Jeffries (*Twitter*, *Blog*)
- Martin Fowler (*Twitter*, *Blog*)
- James Grenning

Scrum

- Ken Schwaber (*Twitter*, *Scrum.org*)
- Mike Beedle
- Jeff Sutherland (*Twitter*, *Scrum.org*)

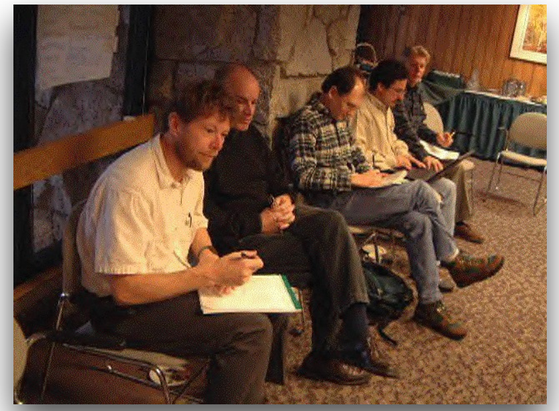
Pragmatic Programmers

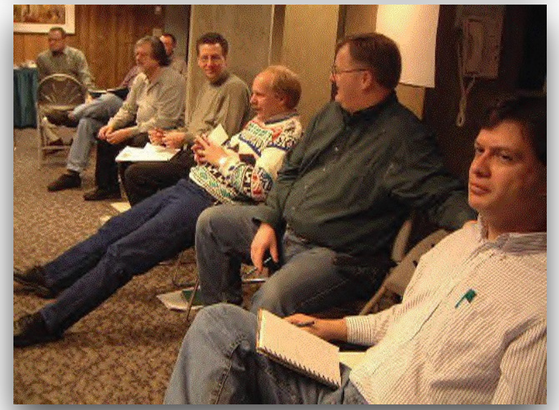
- Andrew Hunt (*Twitter*, *Pragmatic Programmer*)
- Dave Thomas (*Twitter*, *Pragmatic Programmer*)

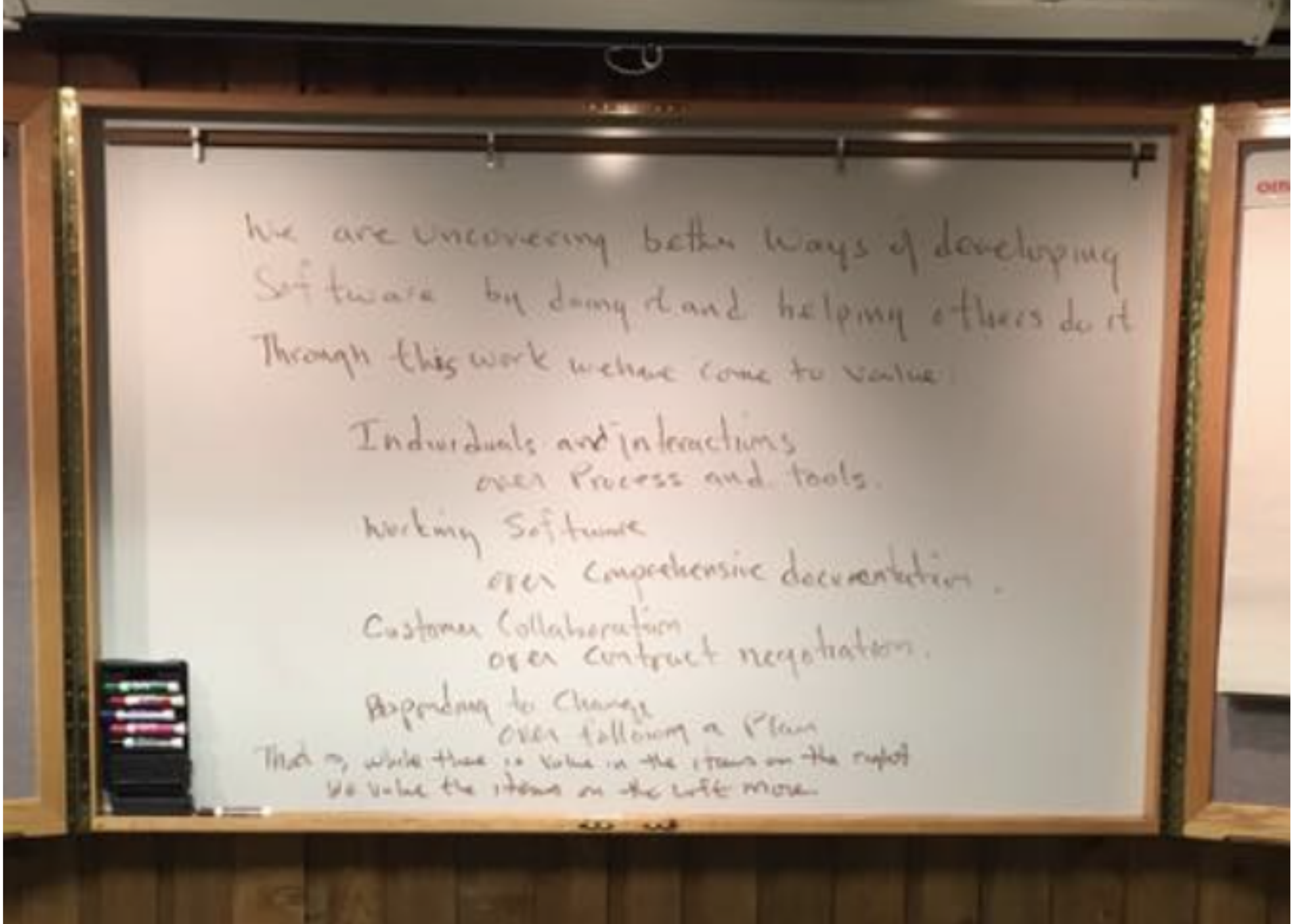
- Alistair Cockburn (*Twitter*, *Blog*)

Crystal

- Jon Kern
- Arie van Bennekum
- Jim Highsmith
- Brian Marick
- Steve Mellor






We are uncovering better ways of developing
Software by doing it and helping others do it
Through this work we have come to value:
Individuals and interactions
over Process and tools.
Working Software
over Comprehensive documentation.
Customer Collaboration
over Contract negotiation.
Responding to Change
over following a Plan
That is, while there is value in the items on the right
We value the items on the left more.

Project Manager's Practices

- Task boards
- Themes
- Story points
- Burn down charts
- Prioritization
- Estimation
- Project velocity

Team Lead's Practices

- Servant leadership
- Information radiator
- Sit together
- Osmotic Communication
- Planning Poker

Valores

Product Owner's Practices

- Backlog item
- User Stories
- Iteration
- Relative ranking
- Product backlog

Developer's Practices

- Refactoring
- Continuous integration
- Pair programming
- Version control system
- Test-driven development

Valores

Individuos e interacciones sobre procesos y herramientas

Valores

Software en funcionamiento sobre documentación exhaustiva

Valores

Colaboración con el cliente sobre negociación de contratos

Valores

Responder al cambio sobre seguir un plan

Principles behind the Agile Manifesto

We follow these principles:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4. Business people and developers must work together daily throughout the project.

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.

10. Simplicity--the art of maximizing the amount of work not done--is essential.

11. The best architectures, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2 Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3 Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

5 Build projects around motivated individuals.
Give them the environment and support they need,
and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

8 Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

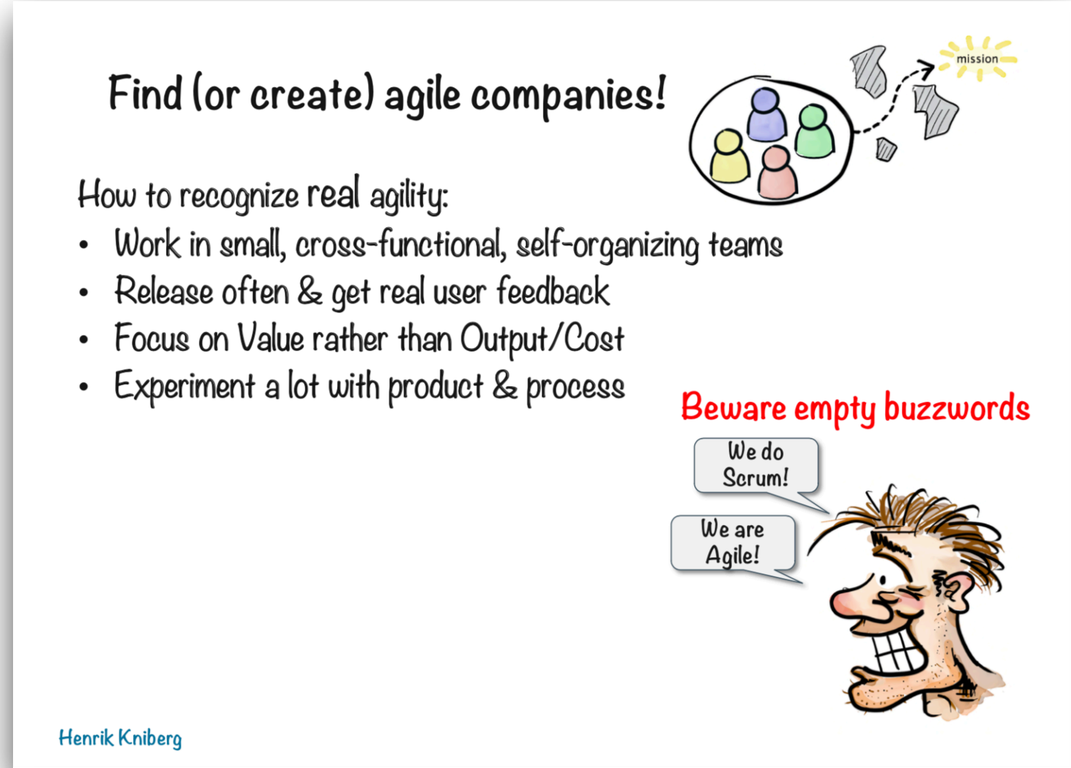
Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Para terminar, dos diapositivas de Henrik Kniberg

- Sacadas de una de sus últimas charlas [Agile intro en KTH]



Para te

- Sacada

www.agilemanifesto.org

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools

Working solutions over comprehensive documentation

Customer collaboration over contract negotiation

Responding to feedback over following a plan

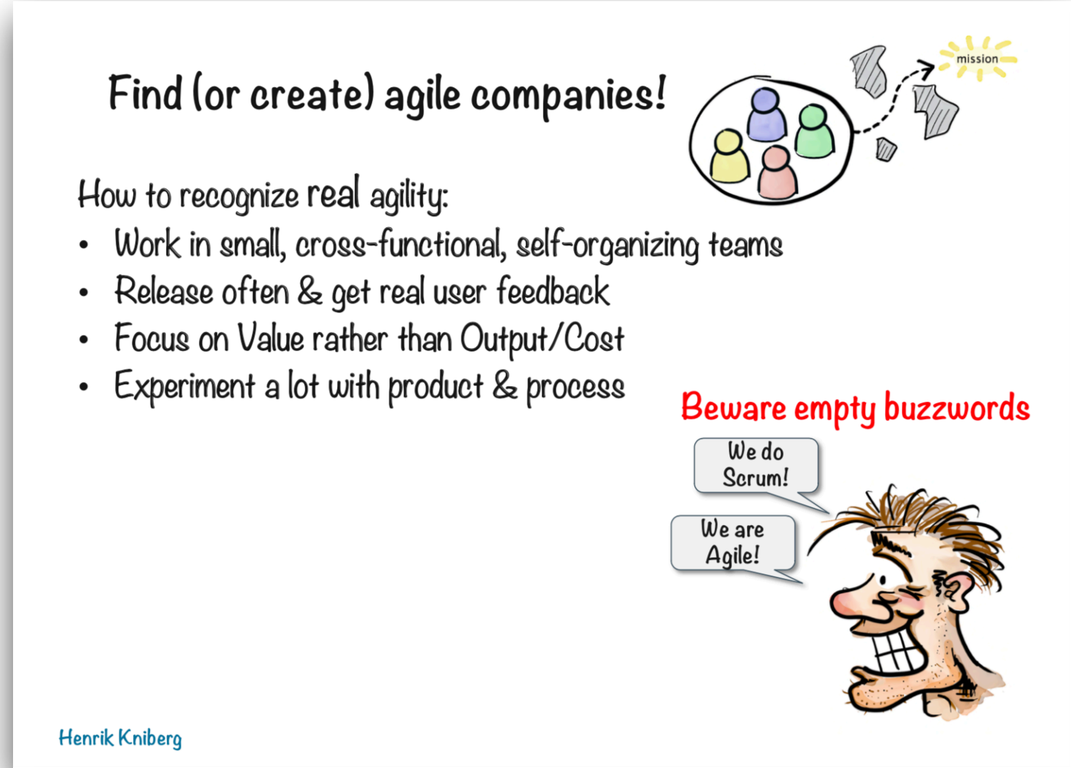
That is, while there is value in the items on
the right, we value the items on the left more.

Henrik Kniberg

uzzwords

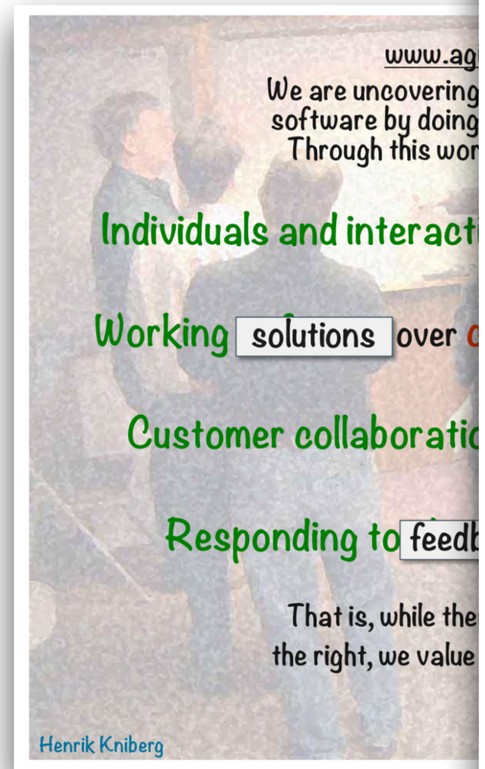
Para terminar, dos diapositivas de Henrik Kniberg

- Sacadas de una de sus últimas charlas [Agile intro en KTH]

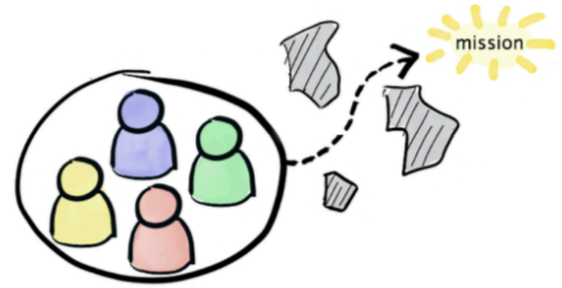


Para termin

- Sacadas de



Find (or create) agile companies!



How to recognize real agility:

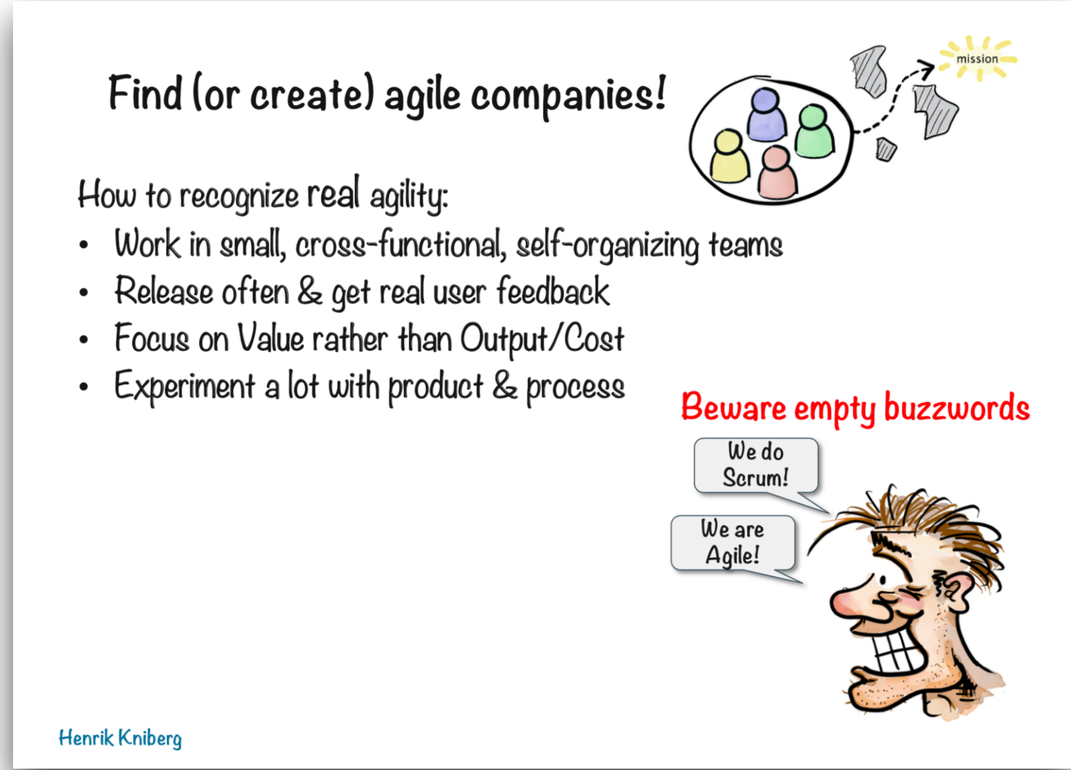
- Work in small, cross-functional, self-organizing teams
- Release often & get real user feedback
- Focus on Value rather than Output/Cost
- Experiment a lot with product & process

Beware empty buzzwords

Henrik Kniberg

Para terminar, dos diapositivas de Henrik Kniberg

- Sacadas de una de sus últimas charlas [Agile intro en KTH]



domingogallardo / **apuntes-mads**

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights

master ▾ **apuntes-mads** / apuntes / manifiesto-agil / **manifiesto-agil.md**

domingogallardo Corrección

1 contributor

784 lines (625 sloc) | 36.1 KB

El Manifiesto Ágil

El Manifiesto Ágil, publicado en febrero de 2001, se considera el documento fundacional de todo el nombre.

Establece un conjunto de valores y principios comunes a un número de ideas y corrientes que fuero críticas con el modelo rígido y pesado de desarrollo de software existente en la época.

En esta sesión vamos a poner en contexto el manifiesto, explicando esas ideas previas, para pasar c contenido del mismo.