

IFIP Working Group 2.16 on Language Design  
February 27, 2012 (Imperial College, London, UK)

(Programming)  
**Design Research and Software Development**  
**Design Thinking . Ideas . Prototypes**

Hasso-Plattner-Institut Potsdam  
Robert Hirschfeld, Bastian Steinert,  
Marcel Taeumel, and Jens Lincke  
hirschfeld@hpi.uni-potsdam.de  
[www.hpi.uni-potsdam.de/swa](http://www.hpi.uni-potsdam.de/swa)

2012-02-27

(Programming)

# Design Research and Software Development

Design Thinking . Research . Prototypes

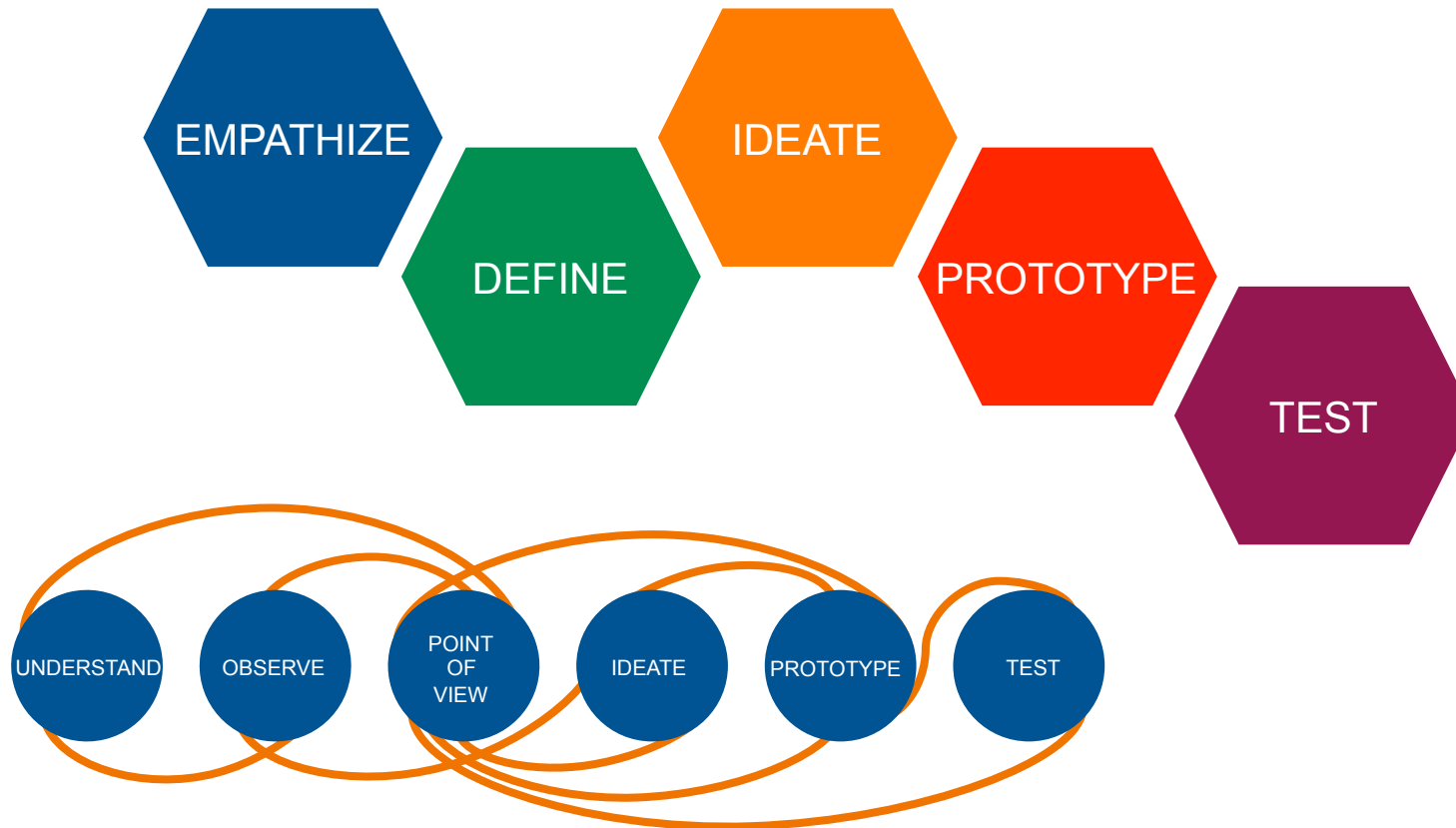


# d.school

<http://bettertastethansorry.com/2010/06/d-school-potsdam/>



# DT Process



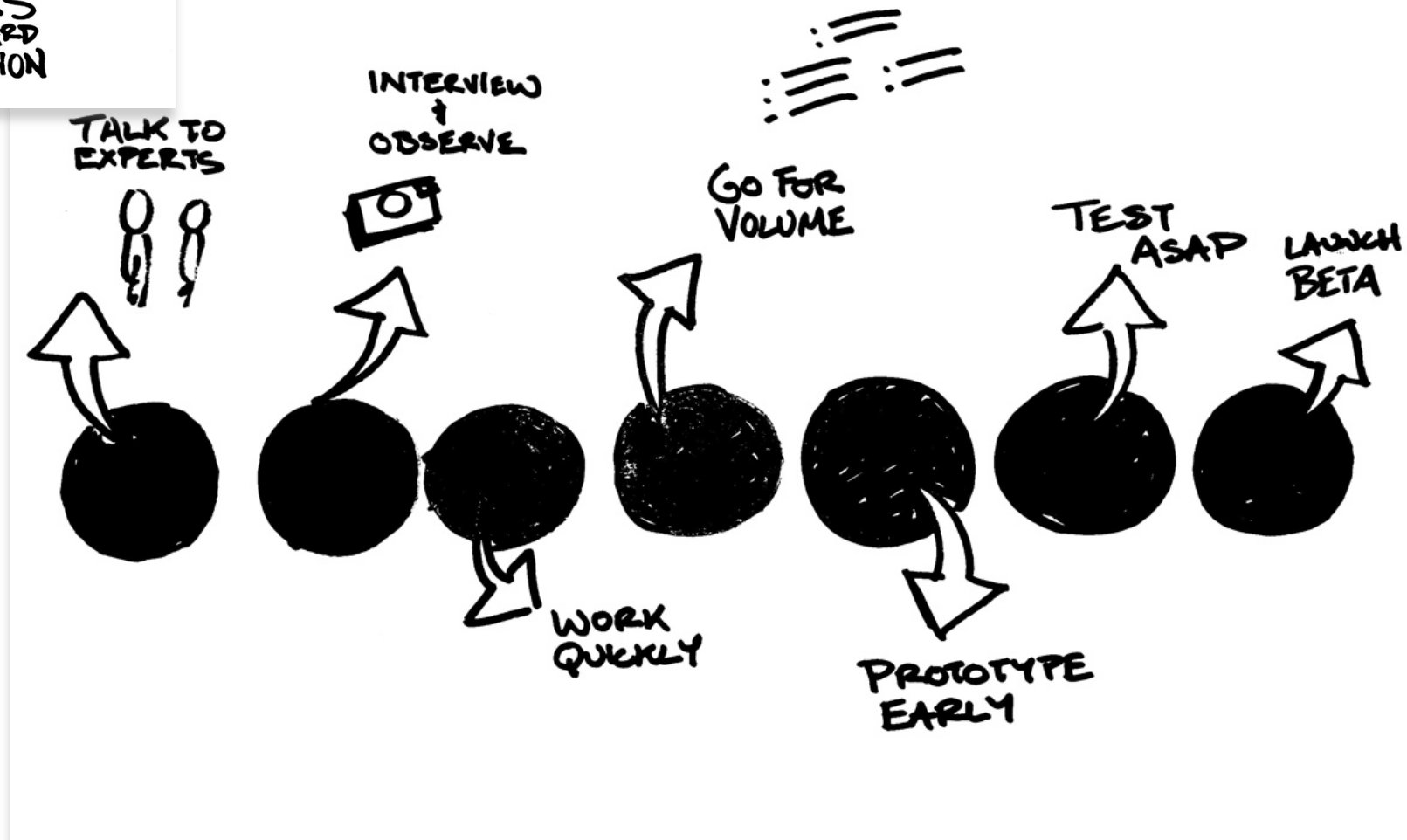


# DT Mindsets



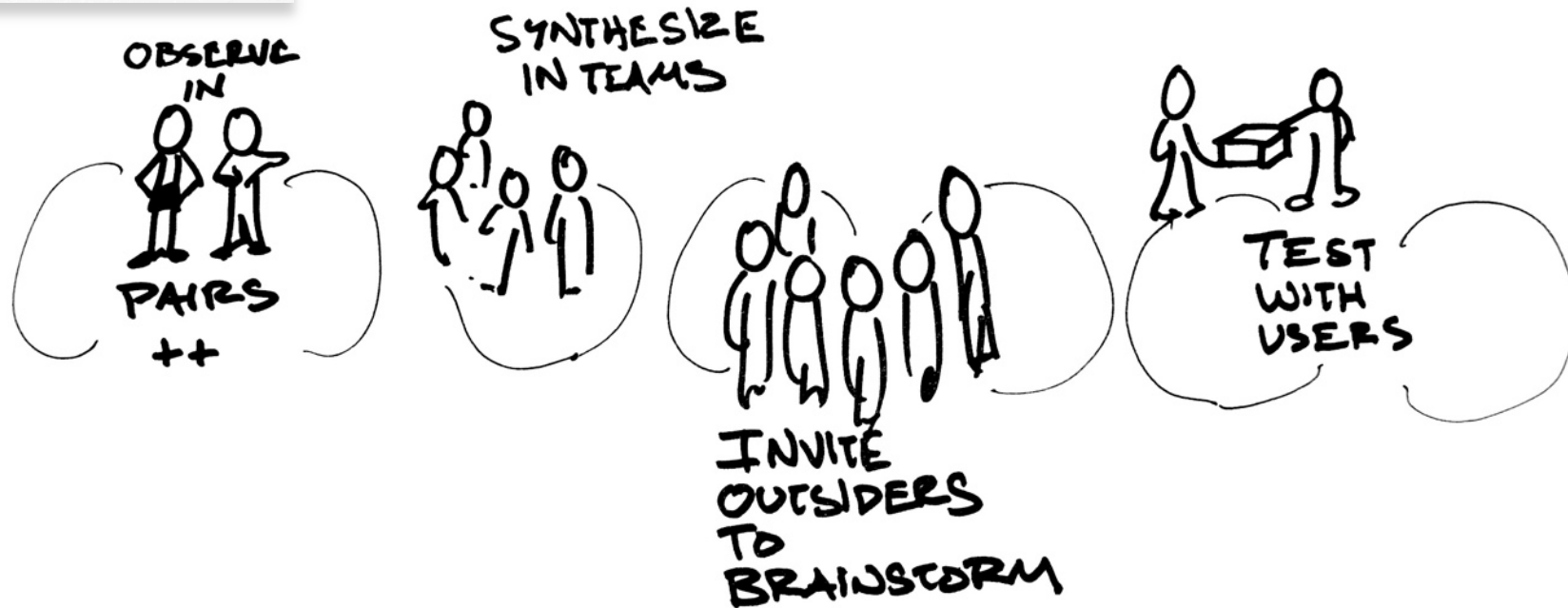
Corey Ford. An Introduction to Design Thinking. dschool 2009  
[https://dschool.stanford.edu/groups/k12/wiki/c739e/attachments/5ba7d/An%20Intro%20to%20Design%20Thinking%2007\\_10\\_09.pdf](https://dschool.stanford.edu/groups/k12/wiki/c739e/attachments/5ba7d/An%20Intro%20to%20Design%20Thinking%2007_10_09.pdf)

# DT Mindsets



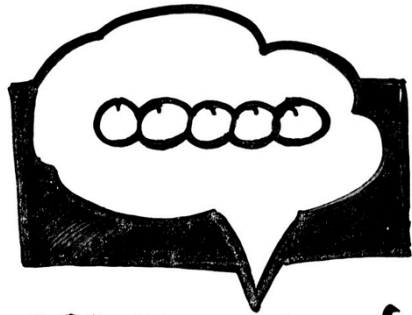
Corey Ford. An Introduction to Design Thinking. dschool 2009  
[https://dschool.stanford.edu/groups/k12/wiki/c739e/attachments/5ba7d/An%20Intro%20to%20Design%20Thinking%202007\\_10\\_09.pdf](https://dschool.stanford.edu/groups/k12/wiki/c739e/attachments/5ba7d/An%20Intro%20to%20Design%20Thinking%202007_10_09.pdf)

# DT Mindsets

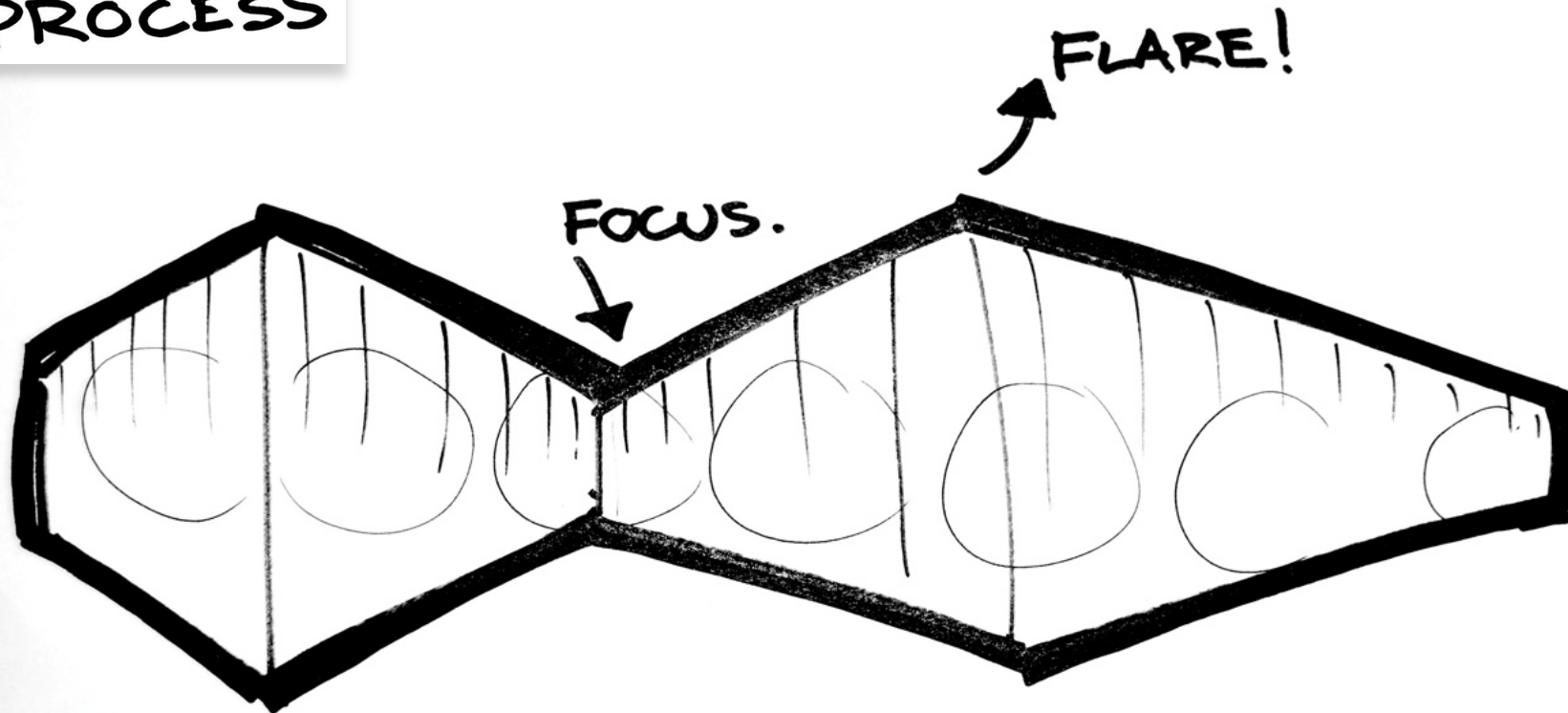


Corey Ford. An Introduction to Design Thinking. dschool 2009  
[https://dschool.stanford.edu/groups/k12/wiki/c739e/attachments/5ba7d/An%20Intro%20to%20Design%20Thinking%202007\\_10\\_09.pdf](https://dschool.stanford.edu/groups/k12/wiki/c739e/attachments/5ba7d/An%20Intro%20to%20Design%20Thinking%202007_10_09.pdf)

# DT Mindsets



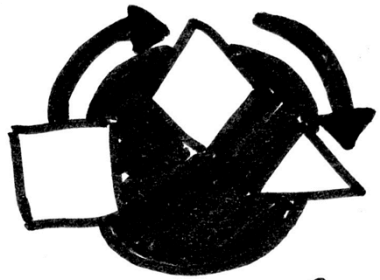
**MINDFUL of  
PROCESS**



Corey Ford. An Introduction to Design Thinking. dschool 2009  
[https://dschool.stanford.edu/groups/k12/wiki/c739e/attachments/5ba7d/An%20Intro%20to%20Design%20Thinking%2007\\_10\\_09.pdf](https://dschool.stanford.edu/groups/k12/wiki/c739e/attachments/5ba7d/An%20Intro%20to%20Design%20Thinking%2007_10_09.pdf)

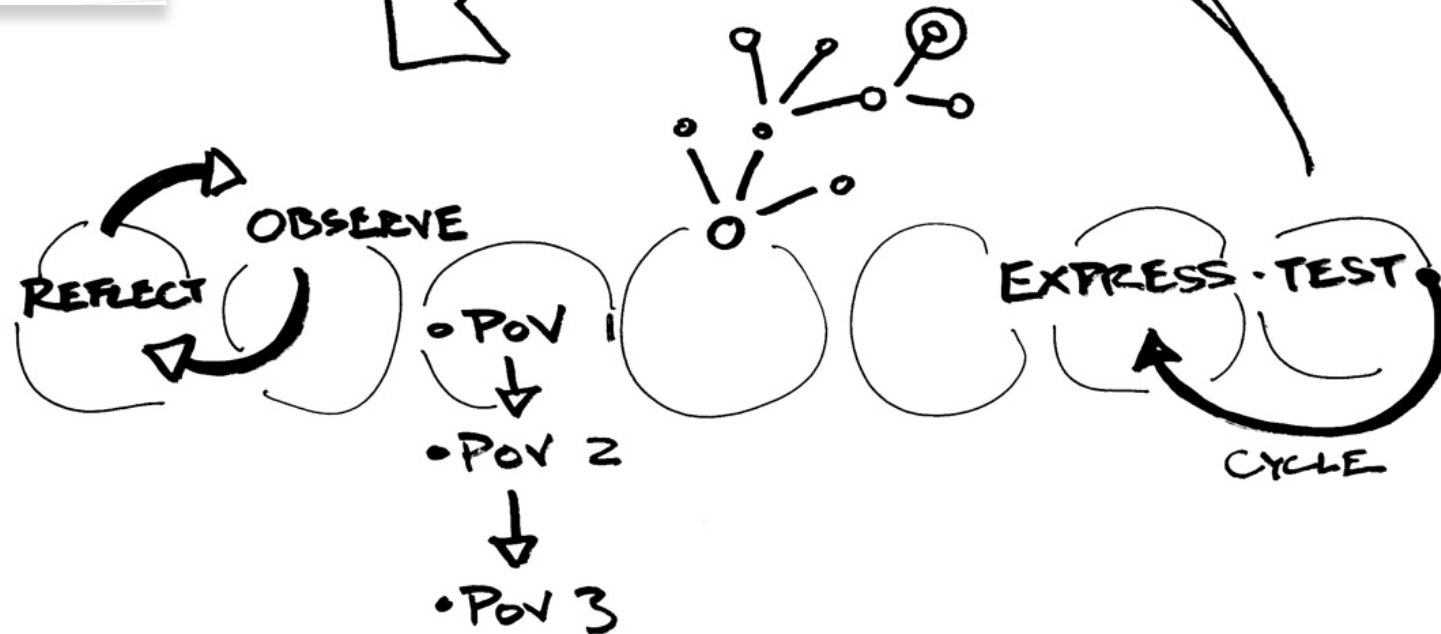


# DT Mindsets



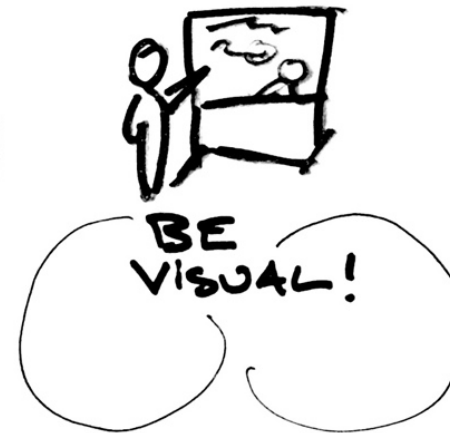
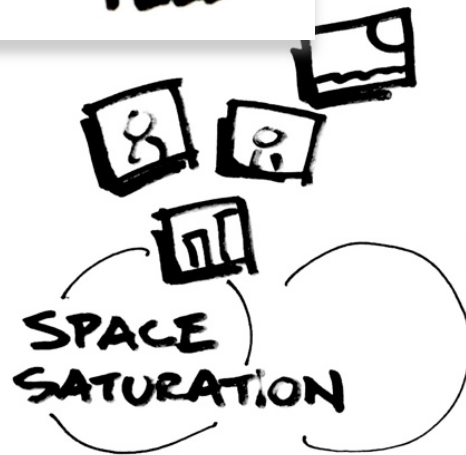
CULTURE OF PROTOTYPING

ITERATE!  
MANY CYCLES



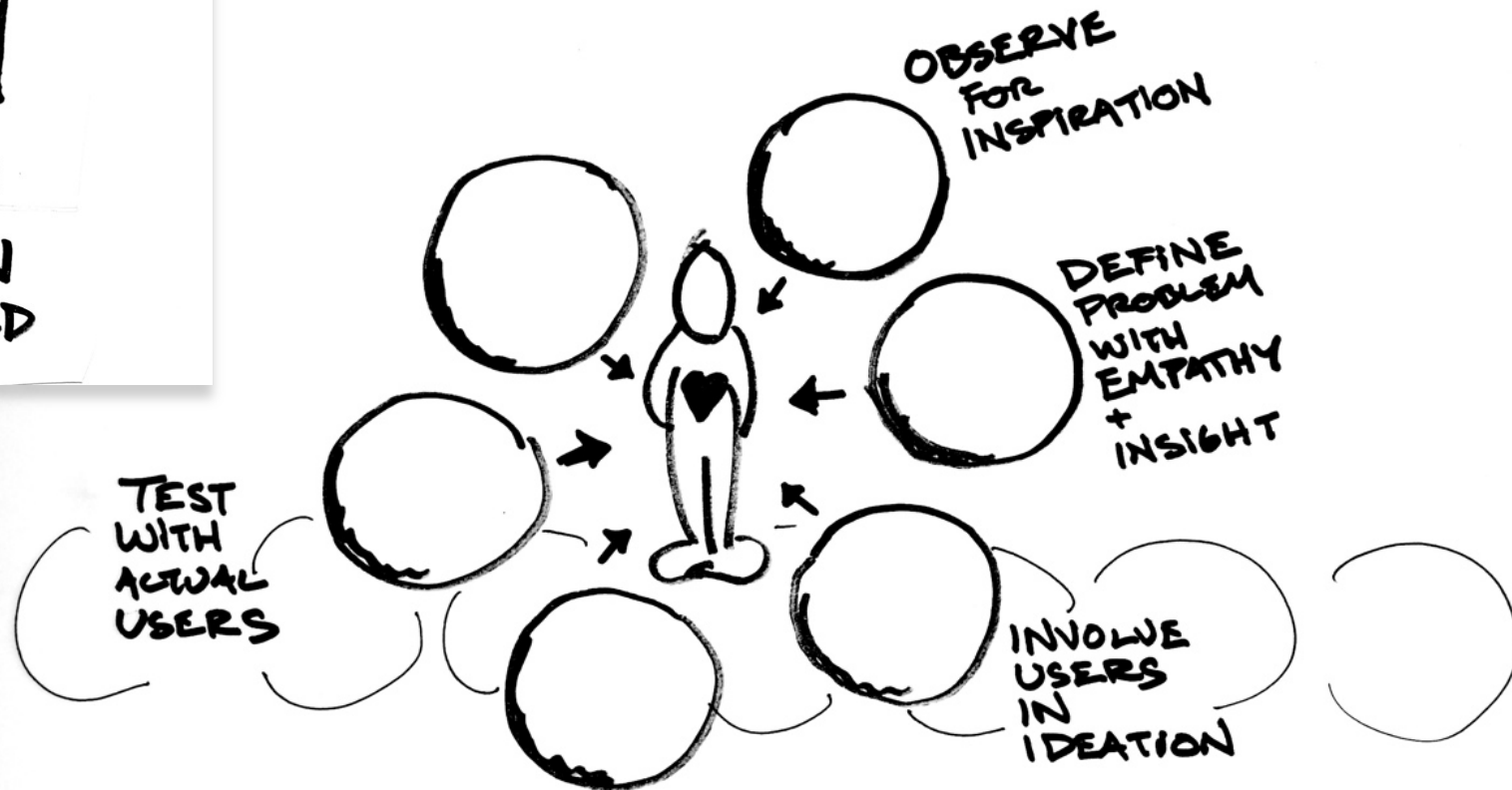
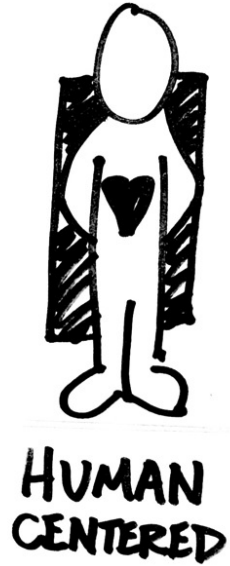
Corey Ford. An Introduction to Design Thinking. dschool 2009  
[https://dschool.stanford.edu/groups/k12/wiki/c739e/attachments/5ba7d/An%20Intro%20to%20Design%20Thinking%2007\\_10\\_09.pdf](https://dschool.stanford.edu/groups/k12/wiki/c739e/attachments/5ba7d/An%20Intro%20to%20Design%20Thinking%2007_10_09.pdf)

# DT Mindsets



Corey Ford. An Introduction to Design Thinking. dschool 2009  
[https://dschool.stanford.edu/groups/k12/wiki/c739e/attachments/5ba7d/An%20Intro%20to%20Design%20Thinking%2007\\_10\\_09.pdf](https://dschool.stanford.edu/groups/k12/wiki/c739e/attachments/5ba7d/An%20Intro%20to%20Design%20Thinking%2007_10_09.pdf)

# DT Mindsets



Corey Ford. An Introduction to Design Thinking. dschool 2009  
[https://dschool.stanford.edu/groups/k12/wiki/c739e/attachments/5ba7d/An%20Intro%20to%20Design%20Thinking%2007\\_10\\_09.pdf](https://dschool.stanford.edu/groups/k12/wiki/c739e/attachments/5ba7d/An%20Intro%20to%20Design%20Thinking%2007_10_09.pdf)

(Programming)

# Design Research and Software Development

Design Thinking . Research . Prototypes



# Design Thinking Research Program

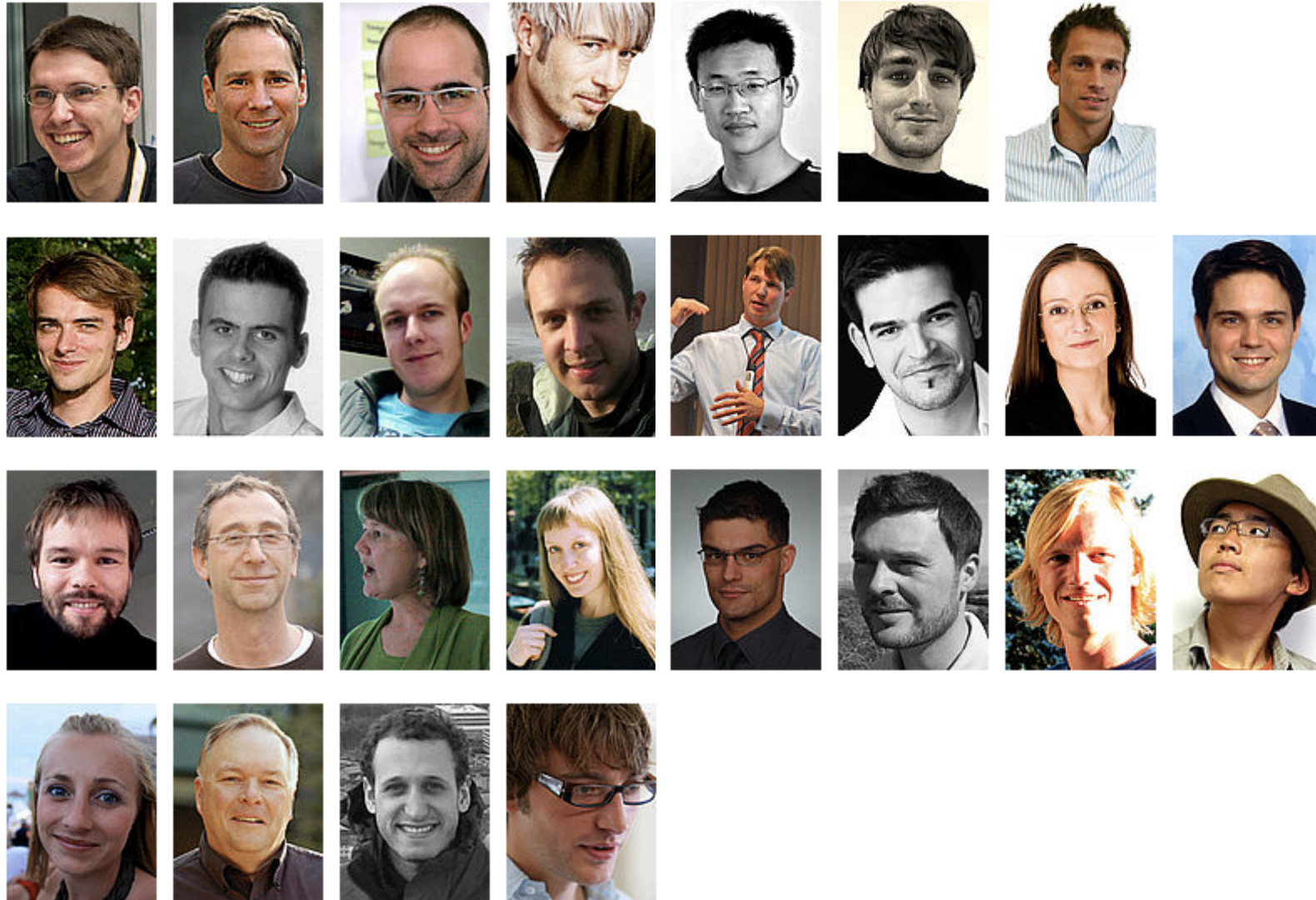




# Design Thinking Research Program



# Design Thinking Research Program

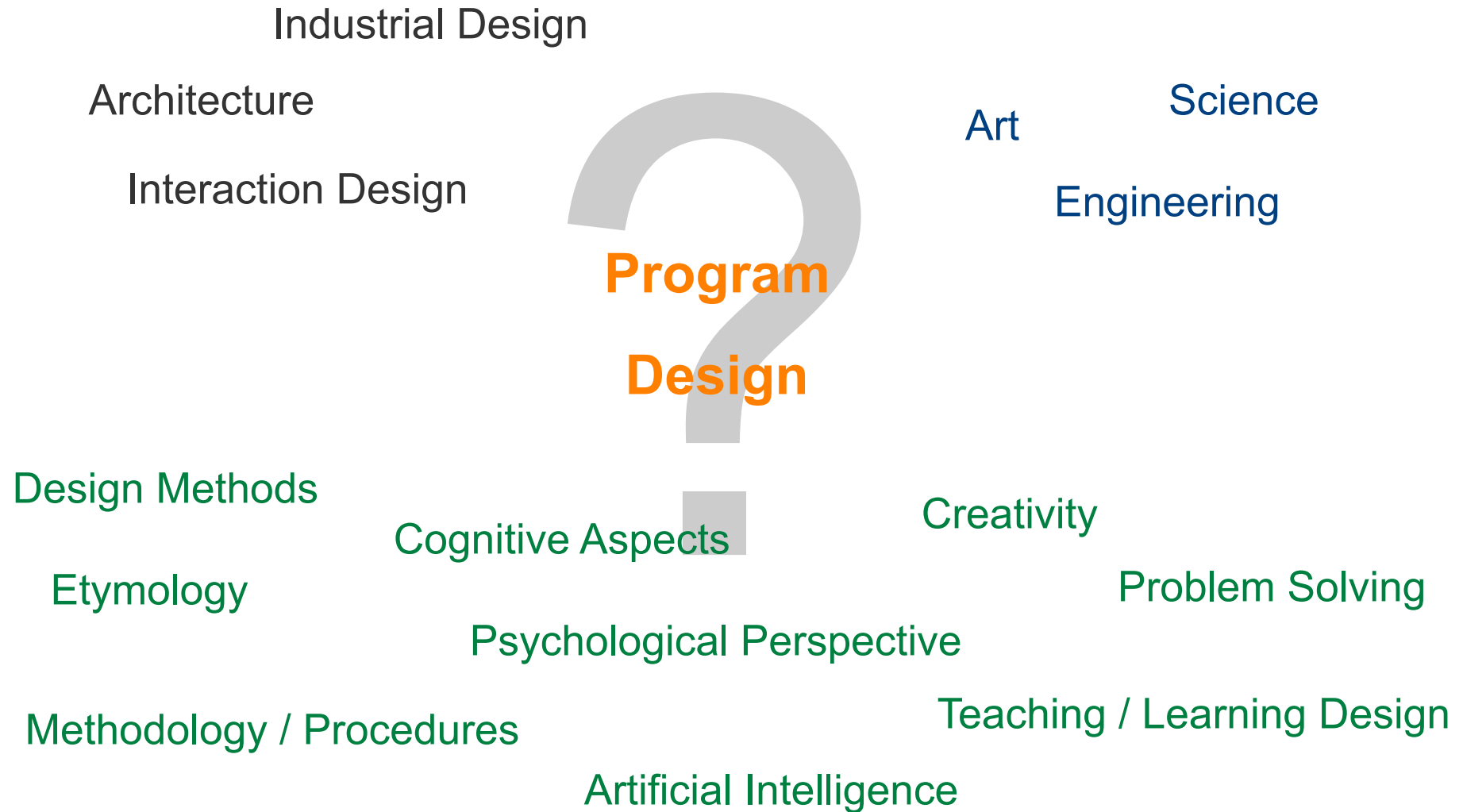




# Design Thinking Research Program

Biomimetics and Dexterous Manipulation Lab (Cutkosky)  
Business Process Technology Group (Weske)  
Center for Design Research, School of Engineering (Leifer)  
Center for Work, Technology, and Organization Management (Hinds)  
d.School, School of Engineering (Roth)  
Enterprise Platform and Integration Concepts (Plattner)  
Human Computer Interaction (Klemmer)  
Internet Technologies and Systems (Meinel)  
Management Science and Engineering (Katila)  
Psychology and Neuroscience (Knutson)  
**Software Architecture Group (Hirschfeld)**  
System Engineering and Modeling Group (Giese)  
...

# Software Architecture Group



Bastian Steinert and Robert Hirschfeld. **Applying Design Knowledge to Programming.** In Hasso Plattner, Christoph Meinel, and Larry Leifer (eds.). *Design Thinking Research: Studying Co-creation in Practice.* pages 259-277, Springer 2011.

# Software Architecture Group

To design “a complex structure, one powerful technique is to discover viable ways of **decomposing** it into semi-independent components corresponding to its many functional parts. The **design of each component can then be carried out with some degree of independence of the design of others**, since each will affect the others largely through its function and independently of the details of the mechanisms that accomplish the function.”, p. 128

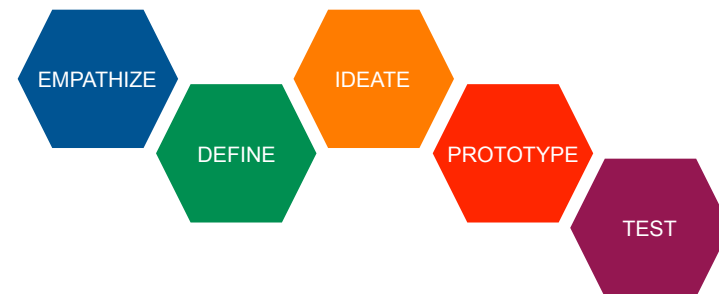
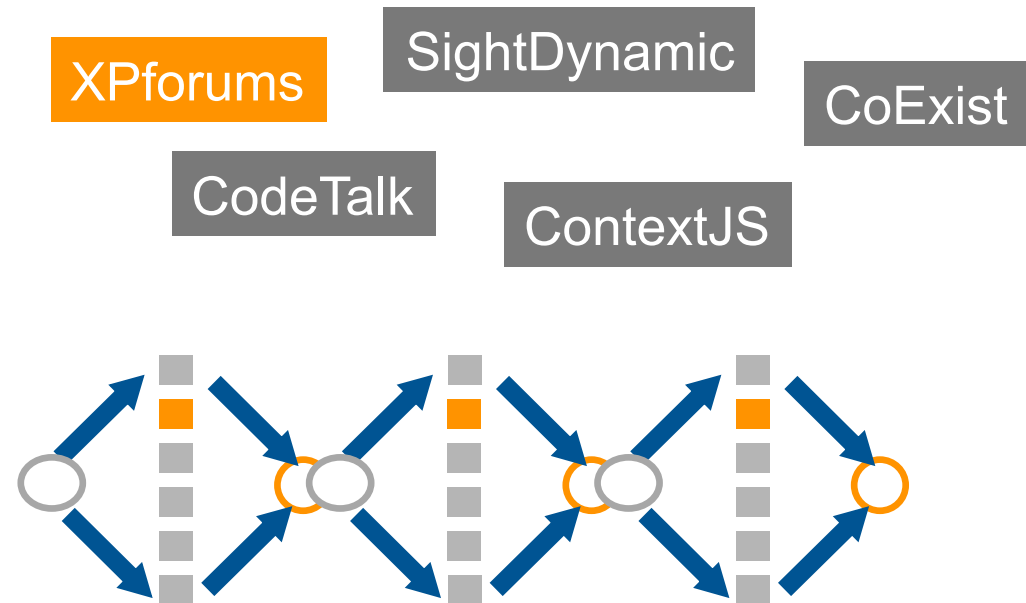
“**Everyone designs who devise courses of action aimed at changing existing situations into preferred ones**. The intellectual activity that produces material artifacts is no different fundamentally from the one that prescribes remedies for a sick patient or the one that devises a new sales plan for a company or a social welfare policy for a state”, p. 130

(Herbert A. Simon. The Science of the Artificial, 3rd edition. MIT Press, 1996)



# Software Architecture Group

- Externalizing thoughts (GTD...)
- Divergent and convergent thinking
- Personas
- Prototypes
- Low resolution
- Tangibility
- Concreteness
- Juxtaposition
- Embrace failure
- ...

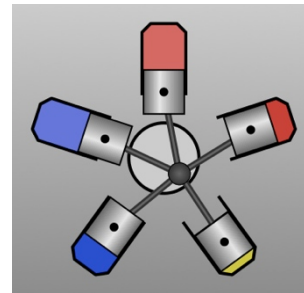


Bastian Steinert and Robert Hirschfeld. **Applying Design Knowledge to Programming**. In Hasso Plattner, Christoph Meinel, and Larry Leifer (eds.). *Design Thinking Research: Studying Co-creation in Practice*. pages 259-277, Springer 2011.

(Programming)

# Design Research and Software Development

Design Thinking . Ideas . Prototypes



(Programming)

# Design Research and Software Development

Design Thinking . Ideas . Prototypes

XPforums

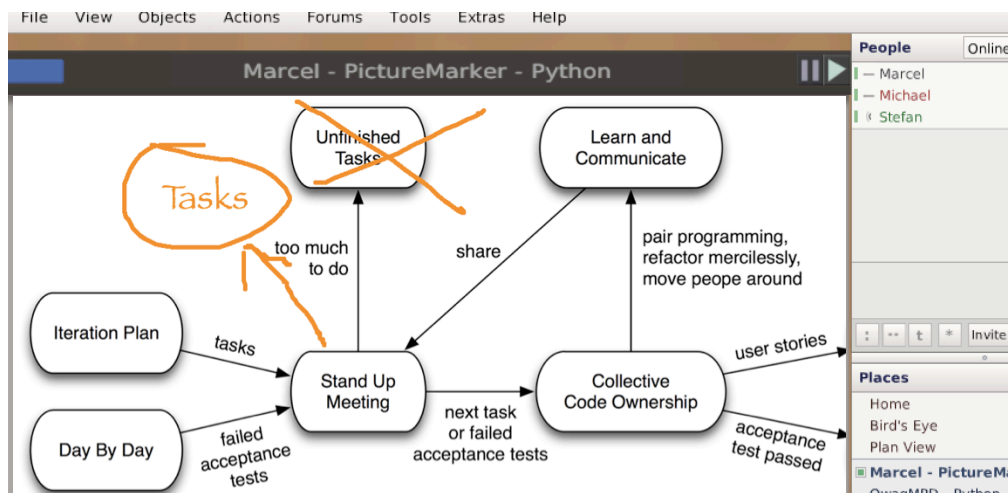
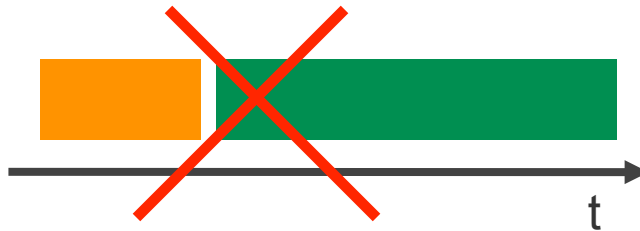
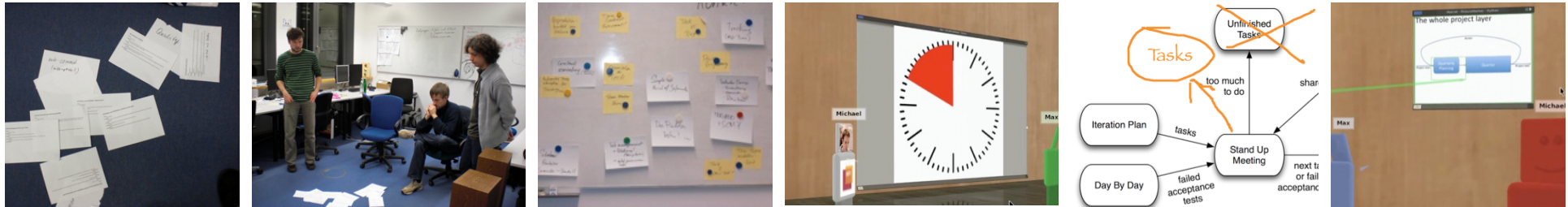
SightDynamic

CoExist

CodeTalk

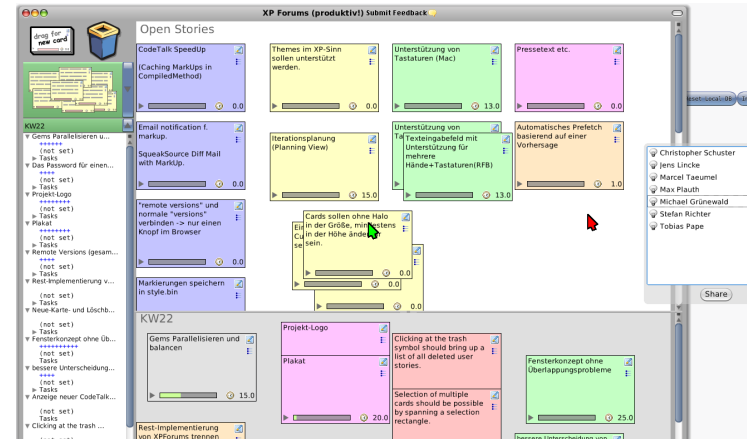
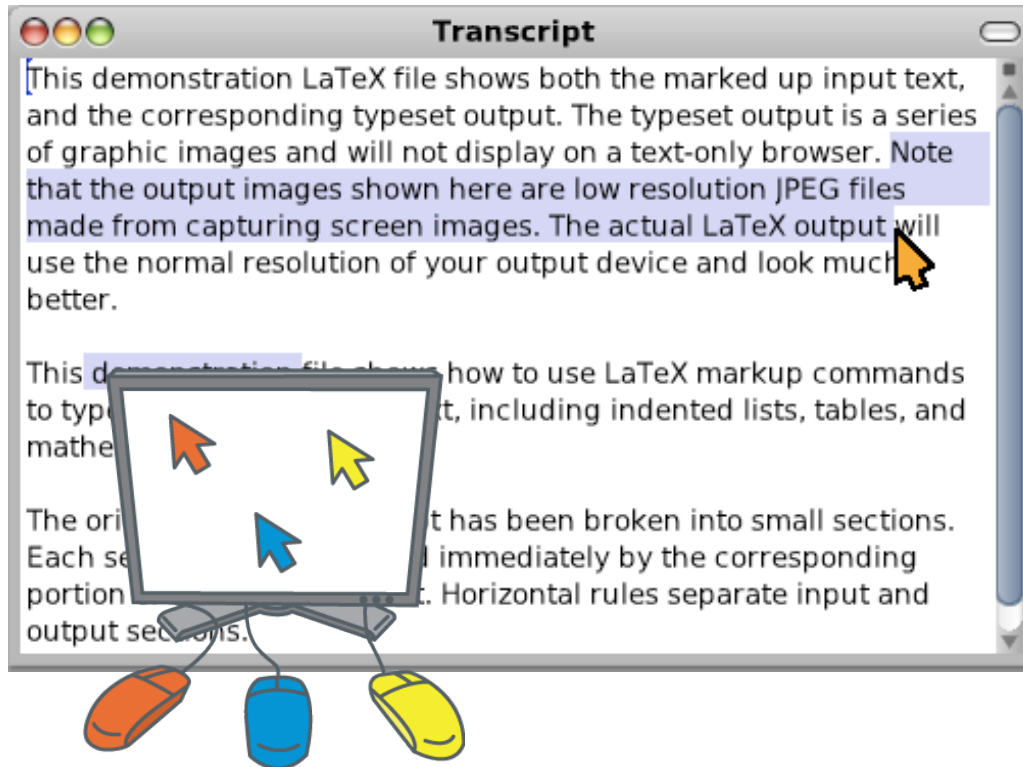
ContextJS

# Design Practices and XP



Robert Hirschfeld, Bastian Steinert, and Jens Lincke. **Agile Software Development in Virtual Collaboration Environments.** In Hasso Plattner, Christoph Meinel, and Larry Leifer (eds.). *Design Thinking: Understand-Improve-Apply.* pages 197-218, Springer 2011.

# Design Practices and XP



Bastian Steinert, Michael Grunewald, Stefan Richter, Jens Lincke, and Robert Hirschfeld. **Multi-user Multi-account Interaction in Groupware Supporting Single-display Collaboration.** In Proceedings of the International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom) 2009, Crystal City, Washington D.C., USA, November 11-14, 2009, IEEE.





(Programming)

# Design Research and Software Development

Design Thinking . Ideas . Prototypes

XPforums

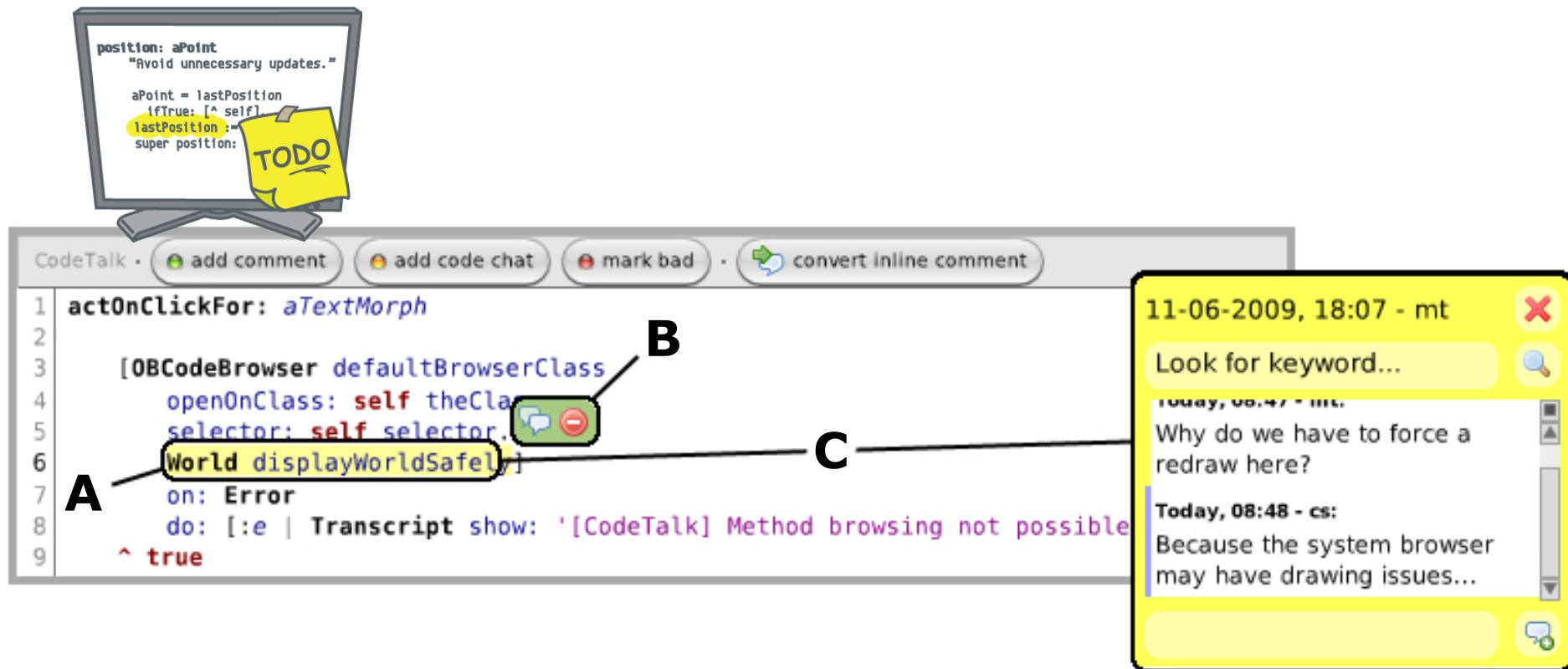
SightDynamic

CoExist

CodeTalk

ContextJS

# Externalizing Thoughts



The diagram illustrates the externalization of thoughts in a code editor. It shows a monitor displaying code with a "TODO" note, a code editor window with annotations A, B, and C, and a chat window showing a discussion about a keyword search.

**Monitor:** Displays code with a "TODO" note. The code includes a comment "Avoid unnecessary updates." and a method call `super position:`.

**Code Editor:** Shows code with annotations A, B, and C. The code includes a method call `World displayWorldSafely` and a comment `Transcript show: '[CodeTalk] Method browsing not possible'`. Annotation A points to the method call, B points to the comment, and C points to the chat window.

**Chat Window:** Shows a discussion about a keyword search. The message reads: "11-06-2009, 18:07 - mt: Look for keyword... Why do we have to force a redraw here? Today, 08:48 - cs: Because the system browser may have drawing issues..."

Bastian Steinert, Marcel Taeumel, Jens Lincke, Tobias Pape, and Robert Hirschfeld.  
**CodeTalk-Conversations About Code.** In Proceedings of the Conference on Creating, Connecting and Collaborating through Computing (C5) 2010, pages 11-18, San Diego, California, USA, January 25-27, 2010, IEEE.

(Programming)

# Design Research and Software Development

Design Thinking . Ideas . Prototypes

XPforums

SightDynamic

CoExist

CodeTalk

ContextJS

# Concrete Data for Abstract Behavior

The screenshot displays the Sight Dynamic IDE 0.2 interface, which is used for analyzing dynamic behavior in software. The interface is divided into several panes:

- Package Viewer (top-left):** Shows a hierarchical view of the project structure. The 'MarbleMania' package is expanded, showing sub-packages like 'initialization' and 'parts bin'. The 'initialization' package is further expanded to show methods like 'createStableState', 'initialize', 'initializeLayout', and 'initializeSubUnits'.
- Call Graph (bottom-left):** Displays a call graph for the test 'MarbleManiaTest >> testStartGame'. The graph shows the sequence of method calls: 'MarbleMania >> initialize', 'MarbleMania >> restart', 'MarbleMania >> board', 'MaMaMarbleBoard >> skipRunningAnimations', 'MarbleMania >> createStableState', 'MarbleMania >> gravity', 'MaMaBoardObserver >> detach', 'MarbleMania >> board', 'MaMaMarbleBoard >> stabilize', 'MarbleMania >> gravity', 'MarbleMania >> board', 'MaMaBoardObserver >> attach', and 'MarbleMania >> pauseMenu'.
- Code Editor (top-right):** Shows the source code for the 'MaMaMarker' class. The 'initialize' method is highlighted, showing the following code:

```
initialize
super initialize.

bounds:=0@0 corner: MaMaMarble defaultWidth@MaMaMarble
defaultHeight + 1.
self
  color: Color transparent;
  borderWidth: 0;
  borderColor: Color black.

hide (function)
  self borderWidth: 0.

handlesMouseDown: (function)
  ^true

mouseDown: (function)
  self owner mouseDown: anEvent.

show (function)
  self borderWidth: 4.
```
- Code Editor (middle-right):** Shows the source code for the 'MarbleMania' class. The 'initialize' method is highlighted, showing the following code:

```
initialize
super initialize.
self
  marker: MaMaMarker new;
  bounds:(0@0 corner: (self class
defaultHeight)@(self class defaultWidth));
  addMorphBack: self marker;
  location: 0@0;
  revive;

animates: anObject
  "Set the value of animates"

  animates := anObject

clickAction (accessing)
  "Answer the value of clickAction"

  ^ clickAction
```
- Code Editor (bottom-right):** Shows the source code for the 'MaMaMarble' class. The 'initialize' method is highlighted, showing the following code:

```
initialize
super initialize.
self
  marker: MaMaMarker new;
  bounds:(0@0 corner: (self class
defaultHeight)@(self class defaultWidth));
  addMorphBack: self marker;
  location: 0@0;
  revive;

animates: anObject
  "Set the value of animates"

  animates := anObject

clickAction (accessing)
  "Answer the value of clickAction"

  ^ clickAction
```

Michael Perscheid, Bastian Steinert, Robert Hirschfeld, Felix Geller, and Michael Haupt. **Immediacy through Interactivity: Online Analysis of Runtime Behavior.** In Proceedings of the 17th Working Conference on Reverse Engineering (WCRE) 2010, pages 77-86, Beverly, Massachusetts, USA, October 13-16, 2010, IEEE.

# Concrete Data for Abstract Behavior

The screenshot displays the Sight Dynamic IDE 0.2 interface with several panels:

- Package Viewer (top left):** Shows a tree view of the project structure. The 'initialization' package is expanded, showing methods like `createStableState`, `initialize`, `initializeLayout`, and `initializeSubUnits`.
- Call Graph (bottom left):** Displays the execution flow for the test `testStartGame`. The call stack includes `MarbleMania>>initialize`, `MarbleMania>>restart`, `MarbleMania>>board`, `MaMaMarbleBoard>>skipRunni`, and `MarbleMania>>createStableSta`.
- MarbleMania (top middle):** Shows the source code for the `MarbleMania` class. The `initialize` method is highlighted, showing the following code:

```
header := anObject
pause
restart
resume (function)
self statistics isTimeOver
iffalse: [self togglePause].
```
- MaMaMarble (bottom middle):** Shows the source code for the `MaMaMarble` class. The `initialize` method is highlighted, showing the following code:

```
super initialize.
self
  marker: MaMaMarker new;
  bounds:(0@0 corner: (self class
defaultHeight)@(self class defaultWidth));
  addMorphBack: self marker;
  location: 0@0;
  revive;
animates: anObject
  "Set the value of animates"
  animates := anObject
clickAction (accessing)
  "Answer the value of clickAction"
  ^ clickAction
```
- a MarbleMania(3951) (top right):** Shows the instance variables for the `a MarbleMania(3951)` object. The `submorphs` list is expanded, showing objects like `a MaMaPauseMenu(2794)`, `an ImageMorph(457)`, `a Morph(3408)`, `a MaMaStatistics`, `a MaMaMarbleBoard(3285)`, `a MaMaSidebar(3385)`, `a MaMaDestroyer`, and `a MaMaGravity`.
- Code Editor (right):** Shows the source code for the `initialize` method of the `MaMaMarble` class, including the `super initialize.` call and the `self` block.



The screenshot displays the Sight Dynamic IDE 0.2 interface, which is used for visualizing the execution of code. The main window is divided into several panes:

- Package Viewer (top left):** Shows the project structure for 'MarbleMania', including packages like 'initialization' and 'Test'.
- Code Editor (center):** Displays the source code for 'MarbleMania' and 'MaMa Marble'. The 'MarbleMania' code includes methods like 'pause', 'restart', and 'resume'. The 'MaMa Marble' code includes 'initialize' and 'clickAction'. Orange dashed lines connect the call graph to the corresponding code lines.
- Call Graph (bottom left):** A tree view showing the sequence of method calls. The path 'MarbleMania >> createStableState' is highlighted in blue, indicating the current execution point.
- Object Inspector (top right):** Shows the state of the current object, 'a MarbleMania(3951)'. It lists various attributes such as 'bounds', 'owner', 'submorphs', 'fullBounds', 'color', 'extension', 'statistics', 'board', 'sidebar', 'header', 'destroyer', 'pausemenu', and 'gravity'. The 'submorphs' array is expanded to show three elements.
- Method Call Log (bottom right):** Shows the sequence of method calls, including 'initialize', 'hide', 'handlesHouse', 'mouseDown', and 'show'.

Orange boxes highlight the 'pause' and 'restart' methods in the code editor, the 'createStableState' method in the call graph, and the 'submorphs' array in the object inspector. A grey box with orange text is overlaid on the bottom right of the code editor, stating: 'embed concrete data to illustrate abstract behavior'.

(Programming)

# Design Research and Software Development

Design Thinking . Ideas . Prototypes

XPforums

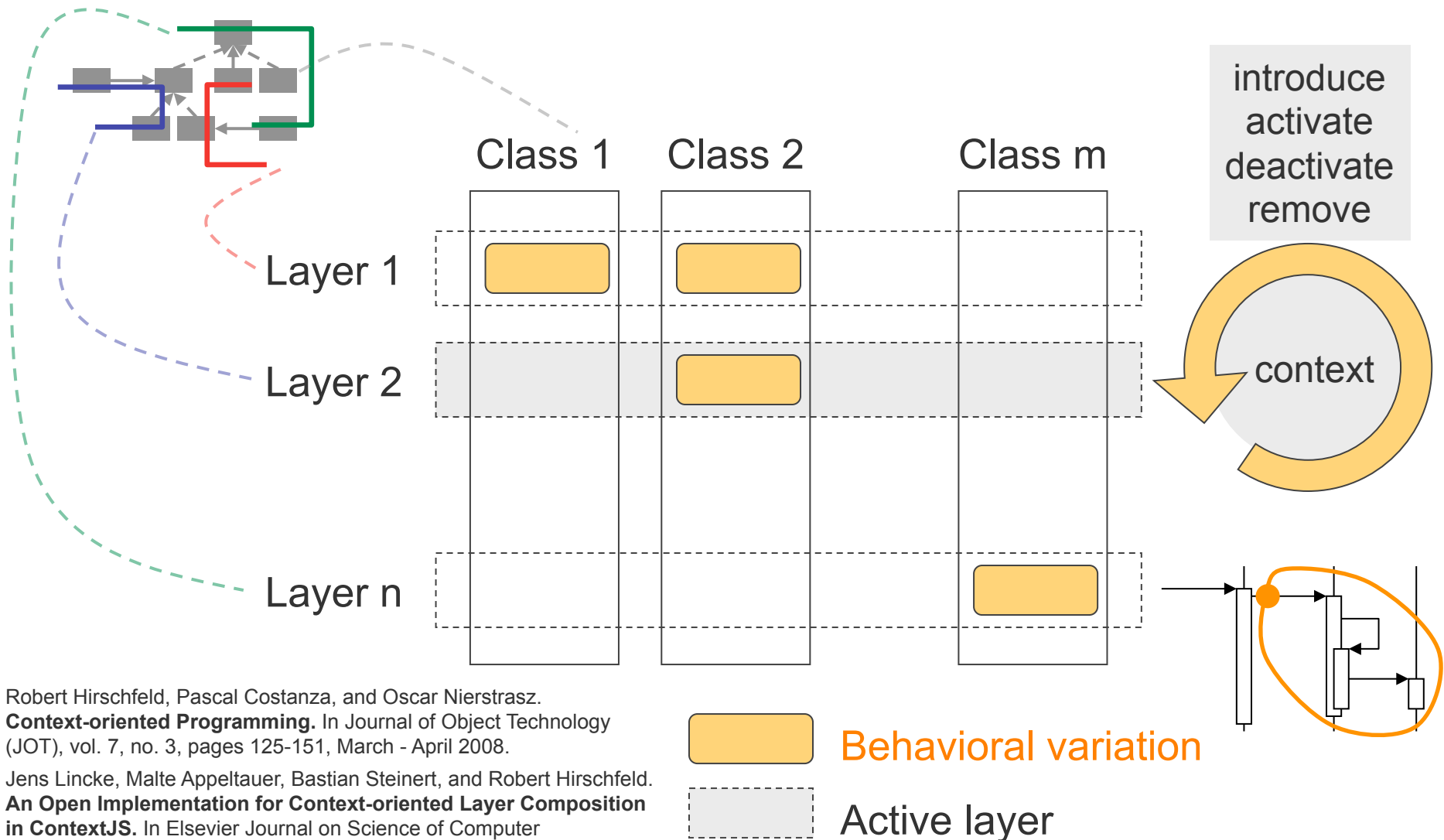
SightDynamic

CoExist

CodeTalk

ContextJS

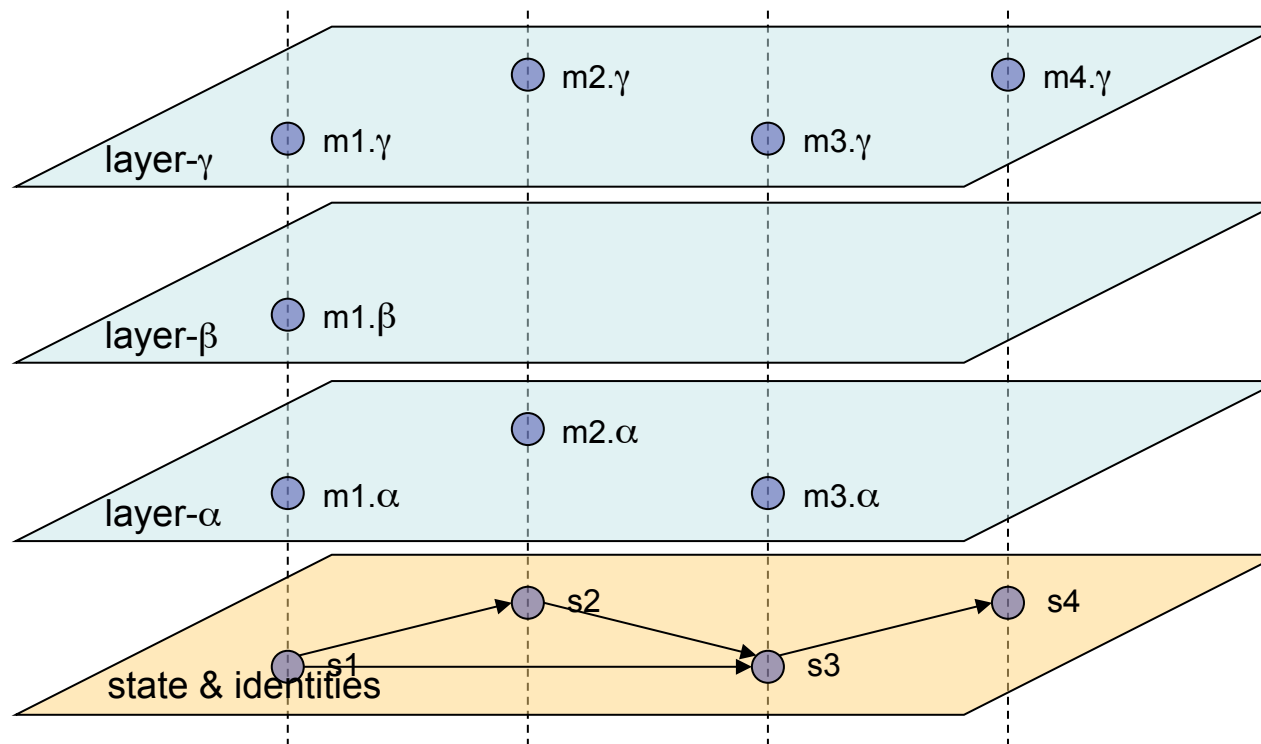
# COP Layers for System Exploration



Robert Hirschfeld, Pascal Costanza, and Oscar Nierstrasz.  
**Context-oriented Programming.** In Journal of Object Technology (JOT), vol. 7, no. 3, pages 125-151, March - April 2008.

Jens Lincke, Malte Appeltauer, Bastian Steinert, and Robert Hirschfeld.  
**An Open Implementation for Context-oriented Layer Composition in ContextJS.** In Elsevier Journal on Science of Computer Programming, Special Issue on Software Evolution, 2011.

# COP Layers for System Exploration



Robert Hirschfeld, Michael Perscheid, and Michael Haupt. **Explicit Use-case Representation in Object-oriented Programming Languages**. In Proceedings of the Dynamic Languages Symposium (DLS) 2011, co-located with the Conference on Object-oriented Programming, Systems, Languages, and Applications (OOPSLA) 2011, pages 51-60, Portland, Oregon, USA, October 24, 2011, ACM DL.

Robert Krahn, Dan Ingalls, Robert Hirschfeld, Jens Lincke, and Krzysztof Palacz. **Lively Wiki - A Development Environment for Creating and Sharing Active Web Content**. In Proceedings of the International Symposium on Wikis and Open Collaboration (WikiSym) 2009, co-located with the Conference on Object-oriented Programming, Systems, Languages, and Applications (OOPSLA) 2009, Disney's Contemporary Resort, Orlando, Florida, USA, October 25-27, 2009, ACM DL.

(Programming)

# Design Research and Software Development

Design Thinking . Ideas . Prototypes

XPforums

SightDynamic

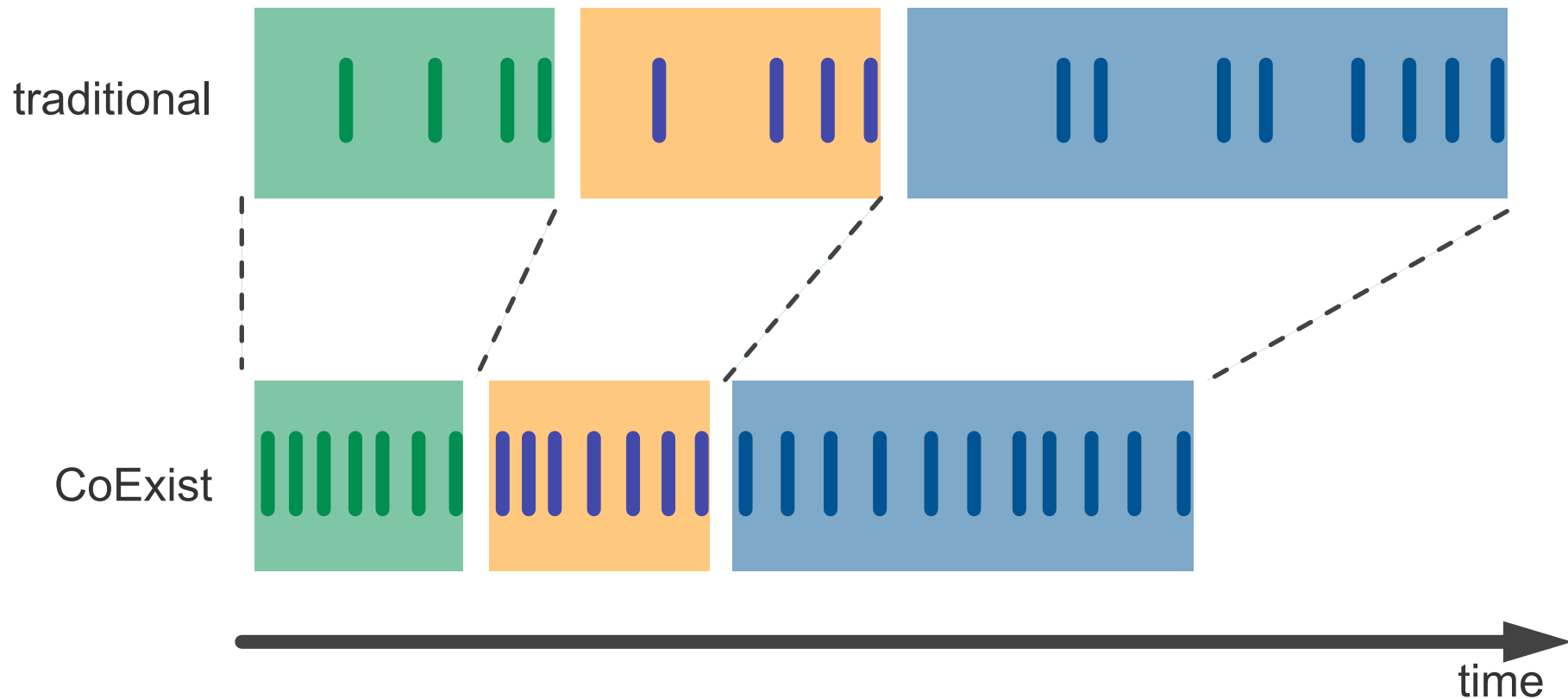
CoExist

CodeTalk

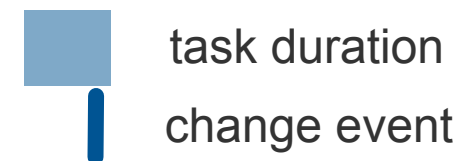
ContextJS



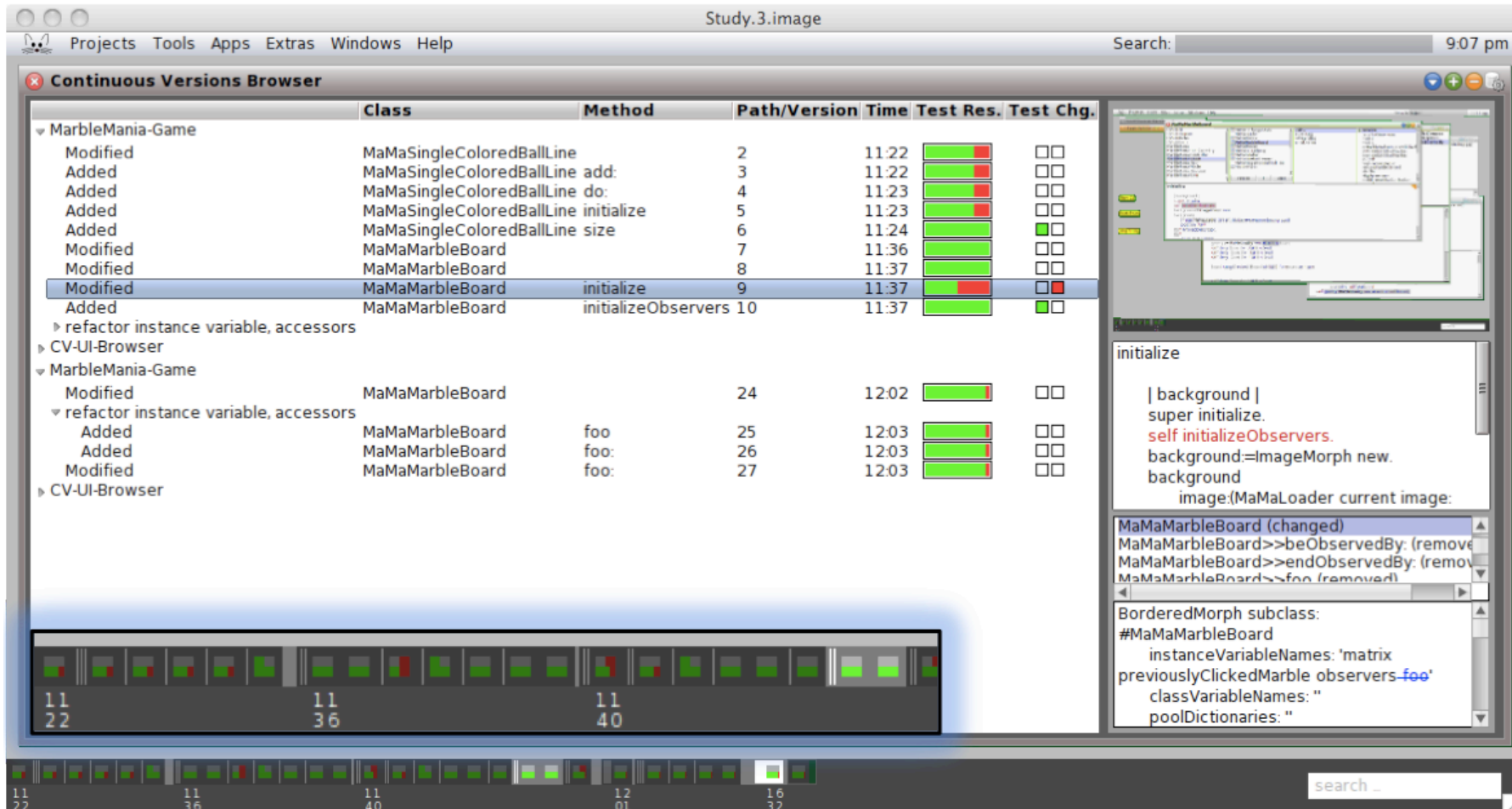
# Defensive Risk Assessments vs. Graceful Failure Recovery



Bastian Steinert, Damien Cassou, and Robert Hirschfeld.  
**The CoExistence of Program Versions Encouraging a  
Trying-out Approach to Program Design.** (submitted...)



# Continuous Versioning



The screenshot shows the 'Continuous Versions Browser' window. The main table lists changes to the 'MaMaMarbleBoard' class, including the 'initialize' method. The right pane shows the source code for the 'initialize' method, which includes a call to 'self initializeObservers'. The bottom of the window features a timeline of changes with version numbers and dates.

	Class	Method	Path/Version	Time	Test Res.	Test Chg.
Modified	MaMaSingleColoredBallLine		2	11:22	<span style="color: green;">■</span> <span style="color: red;">■</span>	<input type="checkbox"/>
Added	MaMaSingleColoredBallLine	add:	3	11:22	<span style="color: green;">■</span> <span style="color: red;">■</span>	<input type="checkbox"/>
Added	MaMaSingleColoredBallLine	do:	4	11:23	<span style="color: green;">■</span> <span style="color: red;">■</span>	<input type="checkbox"/>
Added	MaMaSingleColoredBallLine	initialize	5	11:23	<span style="color: green;">■</span> <span style="color: red;">■</span>	<input type="checkbox"/>
Added	MaMaSingleColoredBallLine	size	6	11:24	<span style="color: green;">■</span> <span style="color: red;">■</span>	<input checked="" type="checkbox"/>
Modified	MaMaMarbleBoard		7	11:36	<span style="color: green;">■</span> <span style="color: red;">■</span>	<input type="checkbox"/>
Modified	MaMaMarbleBoard		8	11:37	<span style="color: green;">■</span> <span style="color: red;">■</span>	<input type="checkbox"/>
Modified	MaMaMarbleBoard	initialize	9	11:37	<span style="color: green;">■</span> <span style="color: red;">■</span>	<input checked="" type="checkbox"/>
Added	MaMaMarbleBoard	initializeObservers	10	11:37	<span style="color: green;">■</span> <span style="color: red;">■</span>	<input checked="" type="checkbox"/>
▸ refactor instance variable, accessors						
▸ CV-UI-Browser						
Modified	MaMaMarbleBoard		24	12:02	<span style="color: green;">■</span> <span style="color: red;">■</span>	<input type="checkbox"/>
▸ refactor instance variable, accessors						
Added	MaMaMarbleBoard	foo	25	12:03	<span style="color: green;">■</span> <span style="color: red;">■</span>	<input type="checkbox"/>
Added	MaMaMarbleBoard	foo:	26	12:03	<span style="color: green;">■</span> <span style="color: red;">■</span>	<input type="checkbox"/>
Modified	MaMaMarbleBoard	foo:	27	12:03	<span style="color: green;">■</span> <span style="color: red;">■</span>	<input type="checkbox"/>
▸ CV-UI-Browser						

```

initialize

| background |
super initialize.
self initializeObservers.
background:=ImageMorph new.
background
image:(MaMaLoader current image:

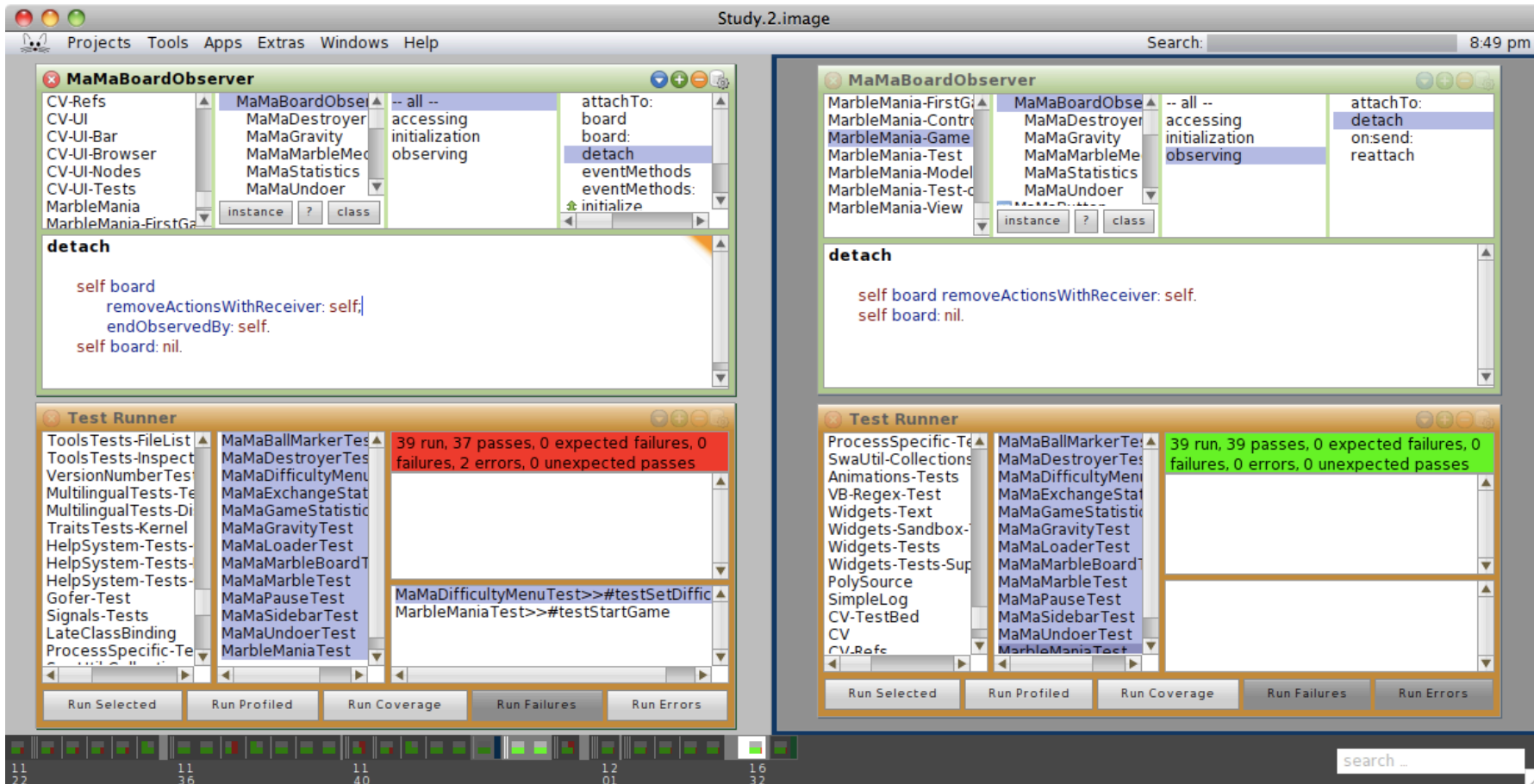
MaMaMarbleBoard (changed)
MaMaMarbleBoard>>beObservedBy: (remove
MaMaMarbleBoard>>endObservedBy: (remov
MaMaMarbleBoard>>foo: (removed)

BorderedMorph subclass:
#MaMaMarbleBoard
instanceVariableNames: 'matrix
previouslyClickedMarble observers-foo'
classVariableNames: "
poolDictionaries: "

```

Bastian Steinert, Damien Cassou, and Robert Hirschfeld.  
**The CoExistence of Program Versions Encouraging a  
 Trying-out Approach to Program Design.** (submitted...)

# Juxtaposing of Both Abstraction and Execution



Bastian Steinert, Damien Cassou, and Robert Hirschfeld.  
**The CoExistence of Program Versions Encouraging a  
 Trying-out Approach to Program Design.** (submitted...)



IFIP Working Group 2.16 on Language Design  
February 27, 2012 (Imperial College, London, UK)

(Programming)  
**Design Research and Software Development**  
**Design Thinking . Ideas . Prototypes**

Hasso-Plattner-Institut Potsdam  
Robert Hirschfeld, Bastian Steinert,  
Marcel Taeumel, and Jens Lincke  
[hirschfeld@hpi.uni-potsdam.de](mailto:hirschfeld@hpi.uni-potsdam.de)  
[www.hpi.uni-potsdam.de/swa](http://www.hpi.uni-potsdam.de/swa)

2012-02-27