



Human-  
Computer  
Interaction  
Institute

Carnegie Mellon University



# Human-Centered Methods for Improving API Usability

**Brad A. Myers**

Human-Computer Interaction Institute

School of Computer Science

Carnegie Mellon University

<http://www.cs.cmu.edu/~bam>

[bam@cs.cmu.edu](mailto:bam@cs.cmu.edu)

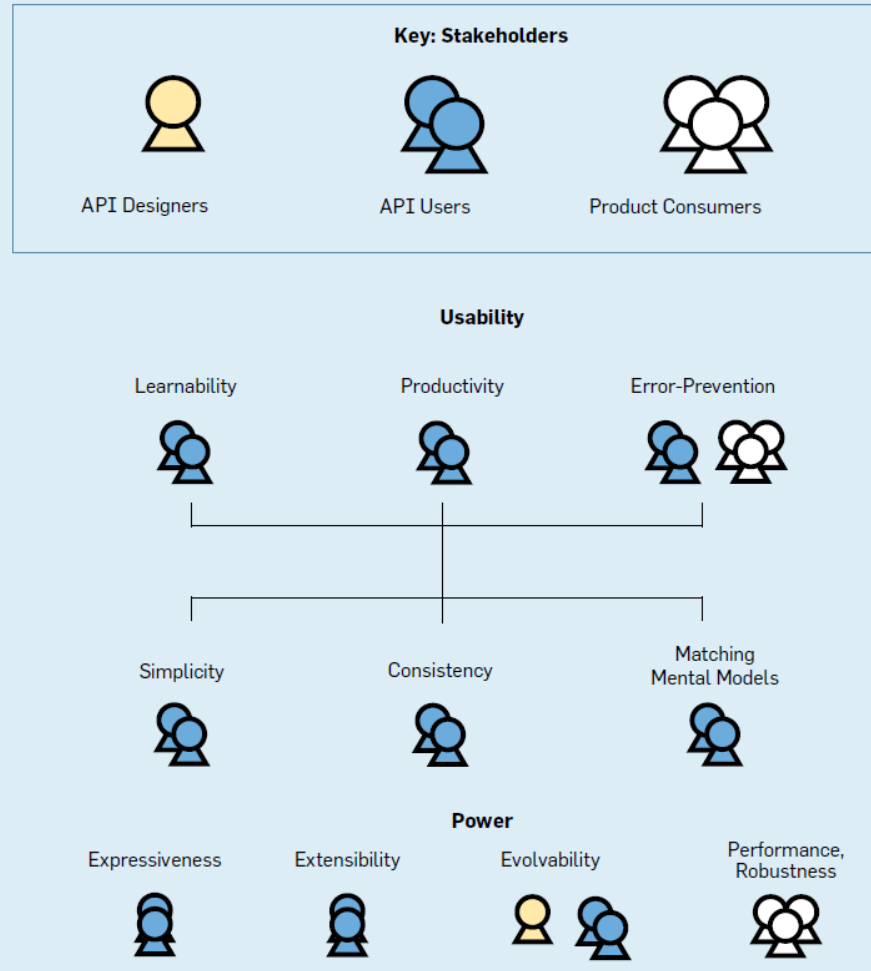
# APIs

- Application Programming Interfaces
- Includes: libraries, toolkits, frameworks, software development kits (SDKs), etc.
- Today: web services, “middleware”
- Also: internal APIs for large software systems
- Provides some functionality for reuse by other developers

# Stakeholders & Their Goals

- [Myers, Stylos, CACM , 2016]

Figure 1. API quality attributes and the stakeholders most affected by each quality.



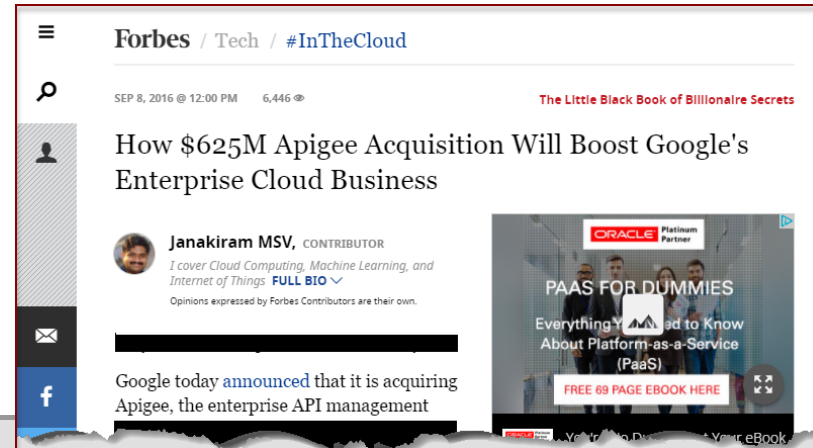
# Why APIs?

- Some design goals for APIs:
  - Information Hiding – hide implementation
  - Provide device independence
  - Enable future changes to low level without requiring changes to application code
  - Protection of critical resources
  - Consistency for product consumer – toolkit can provide commonality
  - More robust code – toolkit implemented correctly
  - Run-time efficiency: provide services in an efficient way
  - Code reuse: Provide useful services only once
  - Programmer Productivity
  - ...

My Goal: Allow API *usability* to be a first-class quality metric considered by API designers

# APIs are Important and Valuable

- [www.programmableweb.com](http://www.programmableweb.com) – 17,508 APIs
- Apigee says 77% of companies rating APIs “important” to making their systems and data available
- Total market for API Web middleware was **\$5.5 billion** in 2014
- Google recently bought Apigee for \$625M



# Why Apply Human-Centered Techniques?

- Programming is a human activity
  - Take the human into account
- “DevX” – like UX for User Experience
- APIs are the “interface” between the programmer and the functionality
- Design should be close to user’s plan
  - “Programming is the process of transforming a mental plan into one that is compatible with the computer.”  
— Jean-Michel Hoc
- Closeness of mapping — Green and Petre
- If an API cannot be used effectively by developers, it doesn’t work!
  - Even if it provides the right functionality
- Using APIs incorrectly has resulted in **bugs and security** problems
- Usability and quality are **key influencers** for the decision about which APIs to use

# “Human Centered Methods” – More Than Just Lab User Studies

- Contextual Inquiry
- Contextual Analysis
- Paper prototypes
- Think-aloud protocols
- Heuristic Evaluation
- Affinity diagrams
- Personas
- Wizard of Oz
- Task analysis
- A/B testing
- Cognitive Walkthrough
- Cognitive Dimensions
- KLM and GOMS (CogTool)
- Video prototyping
- Body storming
- Expert interviews
- Questionnaires
- Surveys
- Interaction Relabeling
- Log analysis
- Storyboards
- Focus groups
- Card sorting
- Diary studies
- Improvisation
- Use cases
- Scenarios
- “Speed Dating”
- ...

# Human Centered Approaches Across the Lifecycle

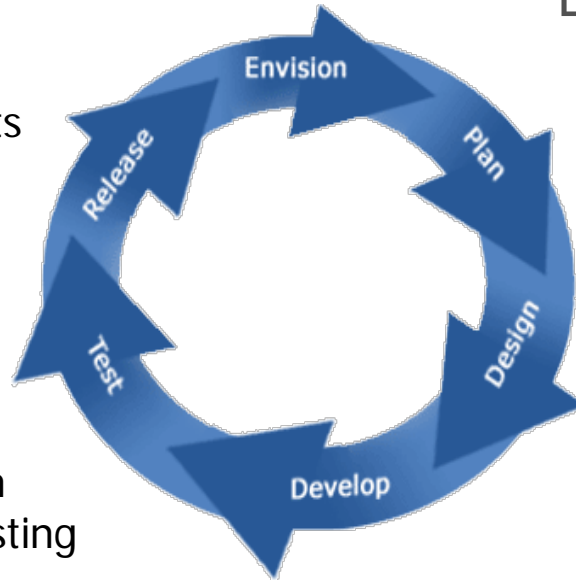
[Myers, Ko, LaToza, Yoon. *IEEE Computer*, 2016]

## Field Studies

- Logs & error reports

## Evaluative Studies

- Expert analyses
- Usability Evaluation
- Formal A/B Lab Testing



## Exploratory Studies

- Contextual Inquiries
- Interviews
- Surveys
- Lab Studies
- Corpus data mining

## Design Practices

- “Natural programming”
- Graphic & Interaction Design
- Prototyping



# HCI Techniques We Have Used for APIs

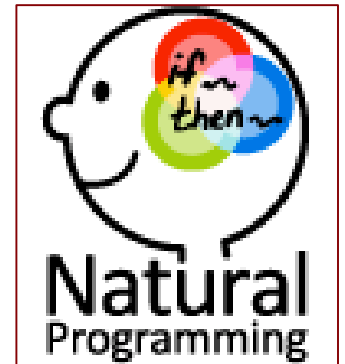
- “Contextual Inquiry” & Field Studies
  - What are the real problems & barriers that developers face?
- “Natural Programming Elicitation”
  - Let programmers express how they expect the functionality to be provided
  - How should this API be designed?
- Expert analyses
  - What are some potential problems with this API?
  - Heuristic Analysis – evaluate based on guidelines
  - Cognitive walkthrough – how hard will this specific task be to learn?
- Lab studies of programmers using an API
  - Does my API work for programmers?
  - What problems do the target developers have with my API?
  - Is *this* design better than *that* one?

# “Natural Programming” Elicitation

- Technique developed by my group to elicit developer’s “natural” expressions
  - Mental models of tasks, vocabulary, etc.
- Blank paper tests
- Must prompt for the tasks in a way that doesn’t bias the answers
- Examples:
  - API Architecture
  - Words used
  - Which methods are on which classes

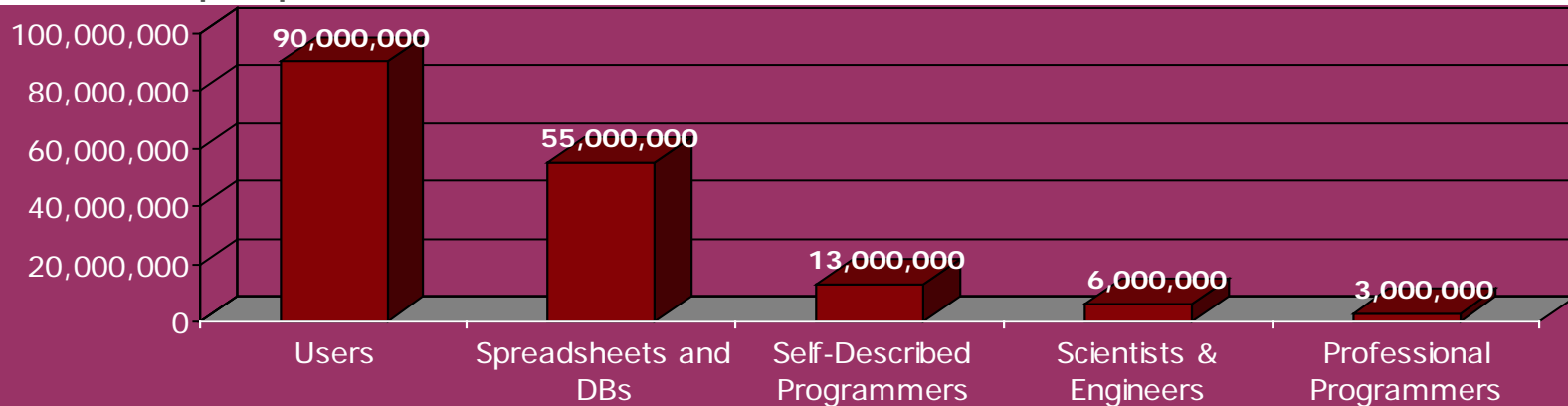
# Context: Natural Programming Project

- Researching better tools for programming since 1978
- Natural Programming project started in 1995
- Make programming easier and more correct by making it more *natural*
  - Closer to the way that people think about algorithms and solving their tasks
- Methodology – human-centered approach
  - Perform *studies* to inform design
    - Provide new knowledge about what people do and think, & barriers
  - Guide the designs from the data
    - Design of programming *languages* and *environments*
  - Iteratively evaluate and improve the tools
- Target novice, expert and end-user programmers

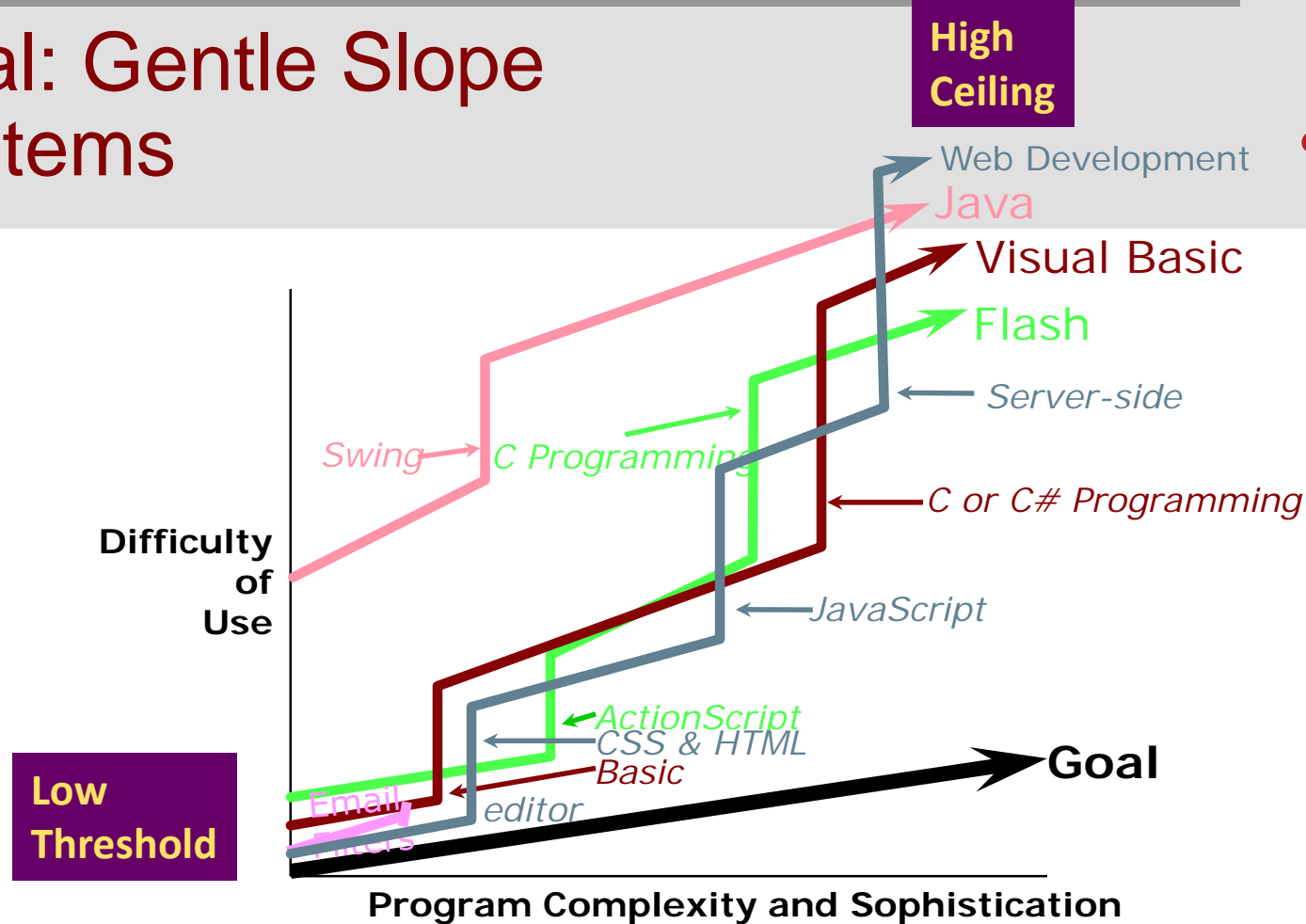


# End User Programming

- People whose primary job is not programming
- [Scaffidi, Shaw and Myers 2005]
  - 90 million computer users at work in US
  - 55 million will use spreadsheets or databases at work (and therefore may potentially program)
  - 13 million will describe themselves as programmers
  - 3 million professional programmers
- All of these people use APIs!



# Goal: Gentle Slope Systems



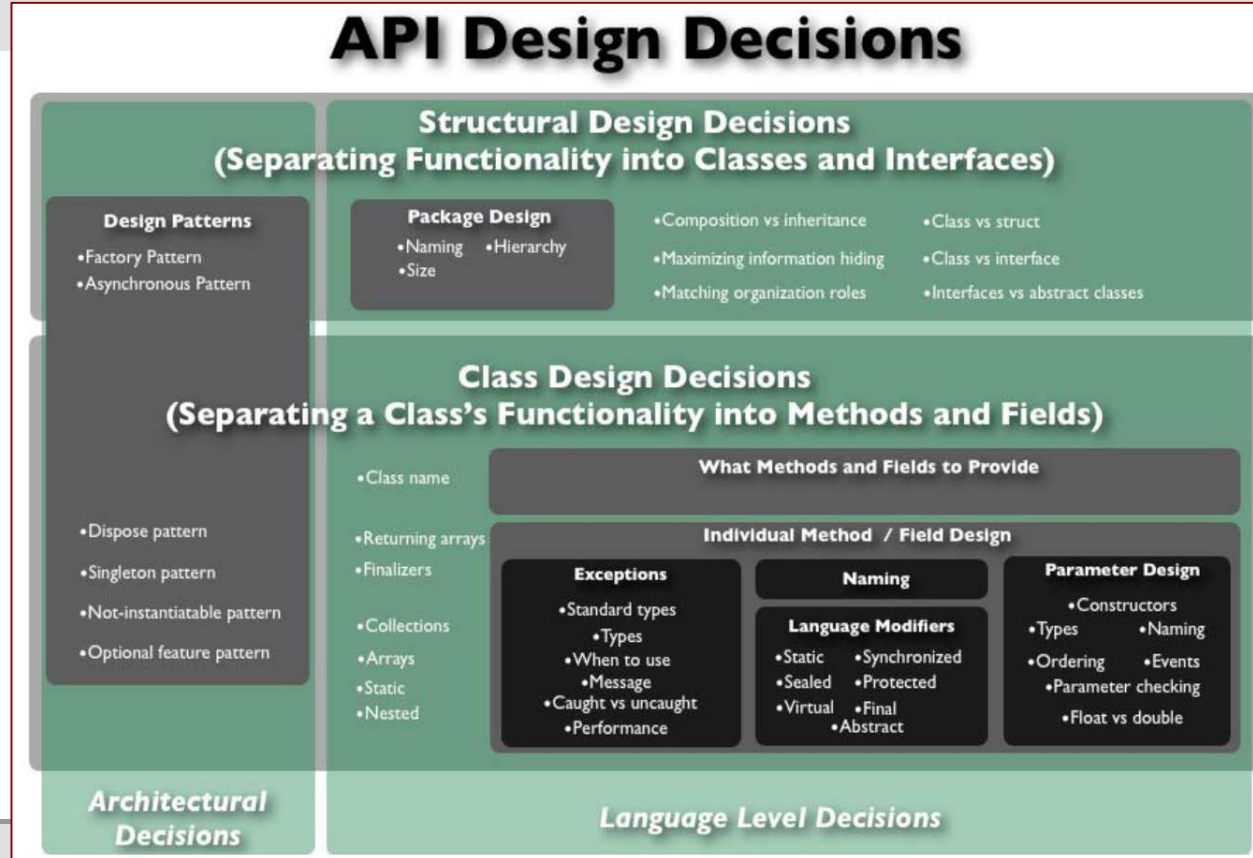
# Our Studies of APIs

- Our work started with Jeff Stylos's PhD, 2005-2009
  - Interned in Microsoft's API Usability group with Steven Clarke, et. al.
- Which programming patterns are most usable?
- Measures: learnability, errors, preferences
- Studied:
  - Required parameters in constructors
  - Factory pattern
  - Object design
  - SAP's APIs
- Tools to help with APIs
- New work: API designers' needs

# What can be addressed?

- All API design decisions
- Tools & documentation for APIs

*[Stylos & Myers, VL/HCC '2007]*



# Required Constructors Study

- Compared **create-set-call** (default constructor)

```
var foo = new FooClass();  
foo.Bar = <get a bar>;  
foo.Use();
```

- vs. required constructors (*immutable* classes):

```
var tempBar = <get a bar>;  
var foo = new FooClass(tempBar);  
foo.Use();
```

- All participants assumed there would be a default constructor
- Required constructors interfered with learning
  - Users wanted to experiment with what kind of object to use first
- Preferred to *not* use temporary variables
- Tradeoff with the security and reliability of **immutable classes**
  - See [Coblentz, Nelson, Aldrich, Myers, Sunshine: “Glacier: Transitive Class Immutability for Java”], Wed @ 11:00



# “Factory” Pattern Study

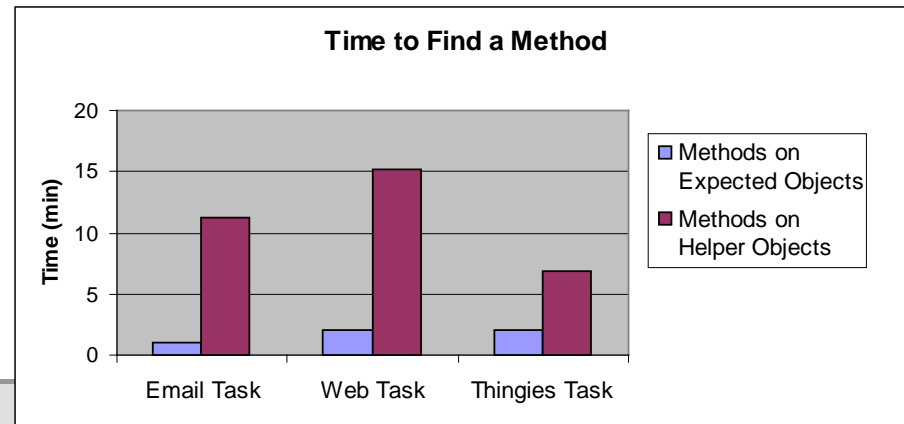
[Ellis. Stylos & Myers, ICSE '2007]

- Instead of “normal” creation: `Widget w = new Widget();`
- Objects must be created by *another* class:

```
AbstractFactory f = AbstractFactory.getDefault();
Widget w = f.createWidget();
```
- Used frequently in Java (>61) and .Net (>13) and SAP
- Results:
  - When asked to design on “blank paper”, **no one** designed a factory
  - Time to develop using factories took **2.1 to 5.3 times longer** compared to regular constructors (20:05 v. 9:31, 7:10 v. 1:20)
  - All subjects had difficulties getting using factories in APIs

# Object Method Placement Study

- Where to put functions when doing object-oriented design of APIs when multiple classes work together
  - `mail_Server.send( mail_Message )`  
vs.  
`mail_Message.send( mail_Server )`
- When desired method is on the class that they start with, users were between **2.4 and 11.2** times faster ( $p < 0.05$ )
- Starting class can be predicted based on user's tasks
- More general terms should be used most commonly
  - `Mail` vs. `Mail_server` class
  - Java `File` class



# Study of APIs for SAP

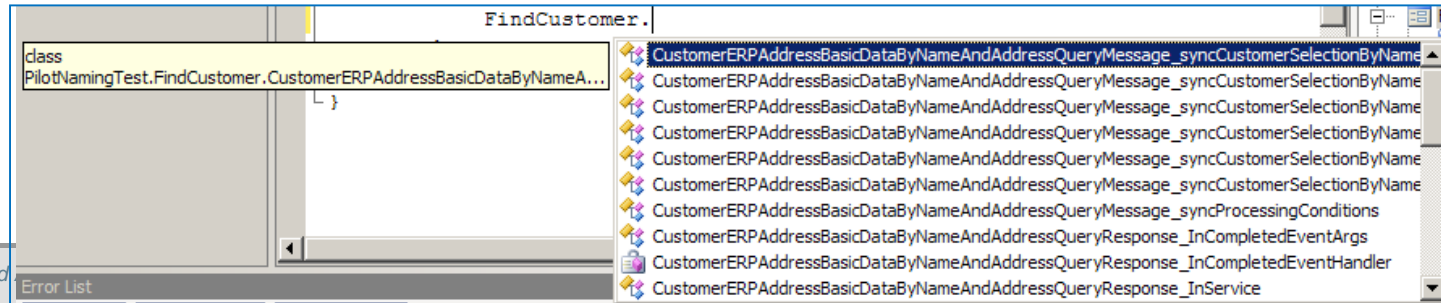
[Jeong, Xie, Beaton, Myers, Stylos, Ehret, Karstens, Efeoglu, Busse, IS-EUD'2009]

- Study APIs for Enterprise Service-Oriented Architectures (“Web Services”)



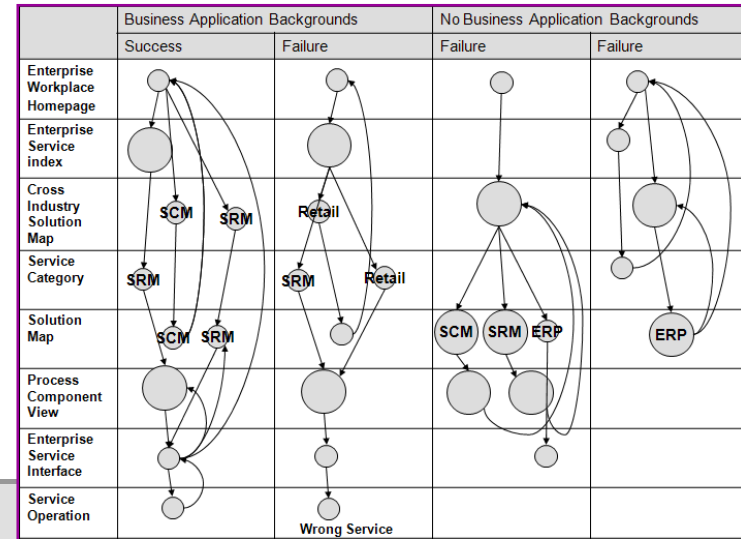
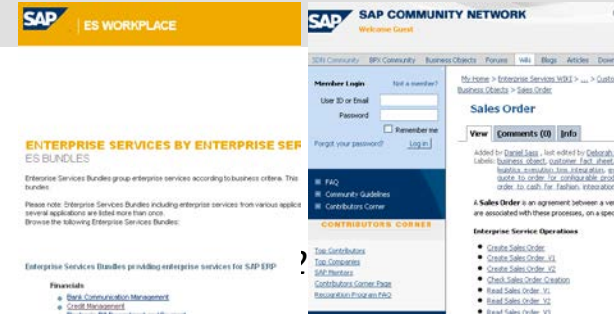
- Naming problems:

- Too long `MaterialSimpleByIDAndDescriptionQueryMessage_syncMaterialSimpleSelectionByIDAndDescriptionSelectionByMaterialDescription`
- Not understandable
- Differences in middle are frequently missed



# eSOA Documentation Results

- Multiple paths: unclear which one to use
- Some paths were dead ends
- Inconsistent look and feel caused immediate abandonment of paths
- Hard to find required information
- Business background helped
- *Many other studies have reported documentation problems*



# Usability study of an API from SAP

*[Stylos, Busse, Graf, Ziegler, Ehret, Karstens, VL/HCC'2008]*

- Jeff Stylos as summer intern at SAP
- SAP “Business Rules Framework Plus” API (BRFplus)
- Interviews with users
  - Identified a mismatch of abstraction level
  - API was very flexible, but users had simple use cases
- Natural programming techniques to identify expected designs
- User studies of redesigned APIs
  - Showed were successful
- Three months total work

# Evaluation based on Guidelines

- Nielsen's Heuristics, Cognitive Dimensions
- Also Cognitive Walkthroughs
- Example: *consistency* violation:

**Code section 1. Two overloadings of the writeStartElement method in Java where localName and namespaceURI are in the opposite order.**

```

void writeStartElement(String namespaceURI,
                    String localName)
void writeStartElement(String prefix,
                    String localName,
                    String namespaceURI)
  
```

*javax.xml.stream.XML.StreamWriter* – [Rama, Kak, 2013]

# SAP's NetWeaver® Gateway Developer Tools

- Plug-in to Visual Studio 2010 for developing using certain SAP APIs
- We used the HCI methods of *heuristic evaluation* and *cognitive walkthroughs* to evaluate early prototypes
- Our recommendations were quickly incorporated due to agile software development process

# Automated Tools (by Others)

- Apply nine metrics to APIs

[Rama, G.M. and Kak, A. Some structural measures of API usability. *Software: Practice and Experience* 45, 1 (Jan. 2013), 75–110]

- E.g., consistency; avoid lists of strings; factory pattern; generic exceptions; ...

- API Concepts Framework takes *uses* of APIs into account

[Scheller, T. and Kuhn, E. Automated measurement of API usability: The API concepts framework. *Information and Software Technology* 61 (May 2015), 145–162]

- Interface Complexity; Implementation Complexity; Setup Complexity



# Our Tools to Help with APIs

- If cannot change API, then fix the documentation and tools

- **Mica**



- **Jadeite**



- **Calcite**



- **Dacite**



- **Euklas**



- **Graphite**



- **Apatite**



# Mica Tool to Help Find Examples



Carnegie Mellon University

[Stylos, Myers VL/HCC'2006]

- **MICA: Makes Interfaces Clear and Accessible**
- Use Google to find relevant pages
- Match pages with Java keywords
- Also notes which pages contain example code or definitions

Mica: Java full screen - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Mail Print

Address http://gem.pebbles.cs.cmu.edu:8080/mica/search?q=full+screen&lang=Java Go

Google Search SafetyBar Block

Mica Java full screen Java Search Advanced

Search Completed [Full-Screen Exclusive Mode API](#)

Do you want to use high-performance graphics in the Java development environment? ... If you've been asking any of these questions, then the **full-screen ...** [java.sun.com/docs/books/tutorial/extra/fullscreen/ - 7k - Cached](#)

[setFullScreenWindow](#)  
[getDefaultScreenDevice](#)  
[GraphicsDevice](#)  
[GraphicsEnvironment](#)  
[isFullScreenSupported](#)  
[BufferCapabilities](#)  
[getDefaultConfiguration](#)  
[BufferStrategy](#)

[Full-Screen Exclusive Mode](#)  
Full-screen exclusive mode is handled through a **java.awt.GraphicsDevice** object. For a list of all available **screen** graphics devices (in single or ... [java.sun.com/docs/books/tutorial/extra/fullscreen/exclusivemode.html - 8k - Cached](#)

[Enabling Full-Screen Mode \(Java Developers Almanac Example\)](#)  
Code Examples from The **Java Developers Almanac** 1.4. [javaalmanac.com/egs/java.awt/screen\\_FullWin.html - 8k - Cached](#)

[Double-Buffering in Full-Screen Mode \(Java Developers Almanac Example\)](#)  
Code Examples from The **Java Developers Almanac** 1.4. [javaalmanac.com/egs/java.awt/screen\\_Flip.html - 10k - Cached](#)

[Office of the President, Republic of China -Taiwan Rainbow Hall](#)  
Taiwan Rainbow Hall: **JAVA** ■■■■■ **JAVA Full Screen** ■■■■■

javascript:rewriteSummaries(1) Internet

# Jadeite: Improved JavaDoc



[Stylos, Faulring, Yang, Myers, VL/HCC'2009]

- **JADEITE: Java API Documentation with Extra Information Tacked-on for Emphasis**

- <http://www.cs.cmu.edu/~jadeite>
- Mine the web for usage of Java APIs
- Fix JavaDoc to help address problems
  - Focus attention on most popular packages and classes using font size
  - “Placeholders” for methods that users want to exist
  - Automatically extracted code examples for how to create classes
  - Related classes

Packages  
[com.sun.mail.dsn](#)  
[com.sun.mail.handlers](#)  
[com.sun.mail.iap](#)  
[com.sun.mail.imap](#)  
[com.sun.mail.imap.protocol](#)  
[com.sun.mail.pop3](#)  
[com.sun.mail.smtp](#)  
[com.sun.mail.util](#)  
**javax.mail**  
[javax.mail.event](#)  
[javax.mail.internet](#)  
[javax.mail.search](#)  
[javax.mail.util](#)

abstract void	<a href="#">saveChanges ()</a> Save any changes made to this message into the message-store when the containing folder is closed, if the message is contained in a folder.
void	<a href="#">send ()</a> Use the Transport.send(message) method to send Messages
protected void	<a href="#">setExpunged (boolean expunged)</a> Sets the expunged flag for this Message.

**See Also (auto-generated):**

[Transport](#)  
[MimeMessage](#)  
[InternetAddress](#)

**Most common way to construct:**

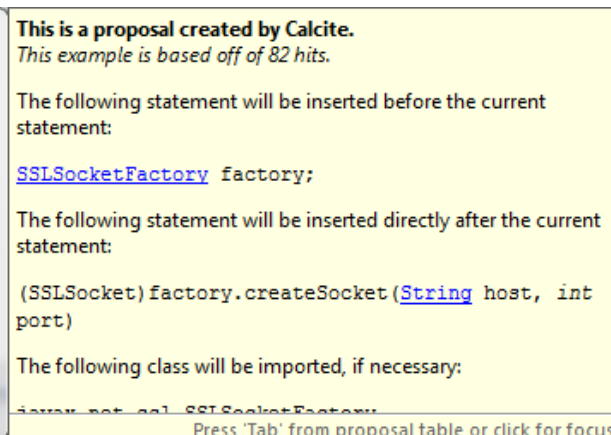
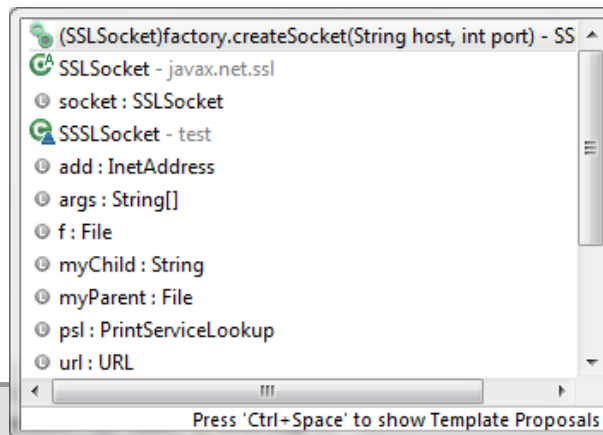
```
SSLSocketFactory factory = ...;  
String host = ...;  
int port = ...;  
SSLSocket socket = (SSLSocket) factory.createSocket(host, port);  
Based on 38 examples
```

# Calcite: Eclipse Plugin for Java



[Mooty, Faulring, Stylos, Myers, VL/HCC'2010]

- **CALCITE: Construction And Language Completion Integrated Throughout Eclipse**
  - <http://www.cs.cmu.edu/~calcite>
- UI = Code completion in Eclipse since familiar and usable
- Code completion in Eclipse augmented with Jadeite's information
  - How to create objects of specific classes?  
`SSLSocket s = | ???`





# Dacite: API Designer Annotates

[Santos, Myers, *Journal of Systems & Software*, April, 2017]

- **DACITE**: Design Annotations for Complementing Interfaces Targeting Effectiveness
- Visiting Professor André L. Santos from University Institute of Lisbon, Portugal
- Use Java annotations to declare properties of APIs
  - Instead of needing to search the web for them
  - More accurate & works for APIs with small user bases
- Processed by Eclipse plugin to help with API discoverability
- Unifies what Calcite did through crawling the web:
  - Supports static factories, factory methods, object builders, helper methods
  - Also adds additional patterns: decorators and composite classes
  - API designers know better what should be annotated
- Lab user study showed effective
  - Twice as many tasks finished

```
public class Collections {
    @StaticFactory public static <T> List<T> emptyList() {}

    public static void sort(@Helper List<?> list) {}
}
```

Design annotations (API designers)

<code>List&lt;String&gt; list =</code>	
<code>Collections.emptyList() : List - Collections</code>	Static factory
<code>List&lt;java.lang.String&gt; - java.util</code>	Returns the empty serializable.
<code>UserCode - test</code>	This example illustrates an empty list:
<code>args : String[]</code>	
<code>main(String[] args) : void - UserCode</code>	

# Euklas: Eclipse Plugin for JavaScript



[Dörner, Faulring, Myers, PLATEAU'2014]

- **EUKLAS: Eclipse Users' Keystrokes Lessened by Attaching from Samples**
  - <http://www.cs.cmu.edu/~euklas>
- Postdoc Christian Dörner
- Brings Java-like analysis to JavaScript
- People often copy from examples in documentation
- Auto-correct uses copy source context for errors due to copy & paste

The screenshot shows the Eclipse IDE with a JavaScript file open. The code defines a function `jawBar(id)` with an `init` method. A copy-paste error is highlighted, and Euklas suggests a correction. The error message at the bottom states: "The function 'init' was not defined!".

```
function jawBar(id) {
  var that = this;
  this.parent = document.getElementById(id);
  this.visible = false;
  this.html = {};
  this.parent.onkeyup = function(e) {
    that.findMatch(e);
  };
  this.init();
}

function init() {
  ...
}
```

Euklas proposes: This code was copied and pasted from the source file 'task\_5\_source.js', and that file contains a function declaration which can be used to fix this problem: `jawBar.prototype.init=function () ...`

[Preview]

```
...
this.visible = false; this.html.div.style.visibility = 'hidden'; this.html.iframe.style.visibility = 'hidden'; };
jawBar.prototype.init=function (){ var that=this;
this.html.div=document.createElement('div'); ...
...
}
```

Problems | @ Javadoc | Declaration

2 errors, 0 warnings, 0 others

Description	Resource	Type
✖ The function 'init' was not defined!	task_5_target.js	/Evaluation line 10

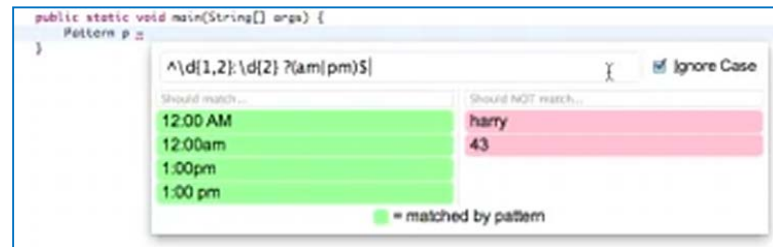
# Graphite: Eclipse Plugin for Literals



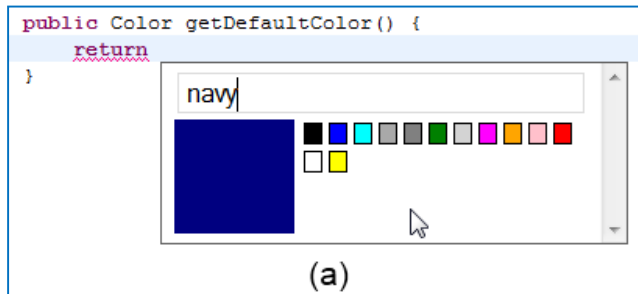
[Omar, Yoon, LaToza, Myers, ICSE'2012]

- **GRAPHITE: GRA**phical **PA**lettes **H**elp **I**stantiate **T**ypes in the **E**ditor.

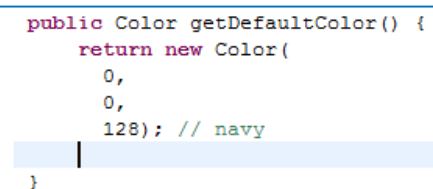
- Pop up a custom palette for specialized constants (literals) in Eclipse



- Regular expressions
- Color palettes
- Customizable



(a)



(b)

# Apatite Documentation Tool



[Eisenberg, Stylos, Faulring, Myers, VL/HCC'10]

- **APATITE: Associative Perusing of APIs That Identifies Targets Easily**
  - <http://www.cs.cmu.edu/~apatite>
- Start with verbs (actions) and properties and find what classes implement them
- Find associated items
  - E.g., classes that are often used together
  - Classes that implement or are used by a method

The screenshot shows two side-by-side search results panels. Both panels have a search bar at the top with the text 'Type here to search...'. The left panel shows results for the search term 'read', with the 'read' method highlighted in green. The right panel shows results for the search term 'write', with the 'write' method highlighted in green. Both panels have a 'FILTER' button on the right side. The results are organized into sections: Packages, Classes, Methods, Actions, and Properties. The 'read' method results include classes like BufferedInputStream, BufferedReader, FileInputStream, and InputStreamReader. The 'write' method results include classes like BufferedInputStream, BufferedWriter, FileOutputStream, and OutputStream.



# New Project

- Funded by a grant from Google
- Interview and survey API designers
  - Processes used
  - Barriers to high usability
  - Information needs about API users
  - Appropriate signals of API usability
- Preliminary discussions with Google, IBM, Amazon, Bloomberg, Microsoft
  - Appear to have quite different processes
  - Different levels of sensitivity to API Usability
- Starting next week!



# Open Challenges

- What other **design patterns** in APIs are problematic or beneficial for usability?
- How to make **coordinating** multiple APIs easier?
- What other design or evaluation **methods** are needed?

# Open Challenges

- Identify **best practices** in API Design
  - How to insure that usability is a **key quality metric** that API designers always consider?
  - What **process** results in the most usable APIs?
    - What kinds of testing should be done on APIs for usability?
    - How should the API design team be organized?
  - What **guidelines** should be followed?
    - Does having guidelines even work?
    - Level of specificity of guidelines?
      - E.g., Smith and Mosier's 1986, 486 pages of guidelines, vs. Nielsen's 10

# A Few Resources

- Brad A. Myers and Jeffrey Stylos, "Improving API Usability", *Communications of the ACM*, vol 59, No. 6, June, 2016, pp. 62-69, [official ACM DL entry](#); [html](#) or [local pdf](#).
- [www.apiusability.org](http://www.apiusability.org)
- <http://www.cs.cmu.edu/~NatProg/apiusability.html>
- <https://www.programmableweb.com/>

# Acronyms are fun!

## And there are lots of Gemstones!!

**Fluorite:**  
Full of  
Low-level  
User



Operations  
Recorded In  
The  
Editor

**Azurite:**

Adding  
Zest to  
Undoing and  
Restoring Improves  
Textual Exploration



**Euklas:**

Eclipse  
Users'  
Keystrokes  
Lessened by  
Attaching from  
Samples



**Variolite:**

Variations  
Augment  
Real  
Iterative  
Outcomes  
Letting  
Information  
Transcend  
Exploration



**Euclase:**

End  
User  
Centered  
Language,  
APIs  
System and  
Environment



**Apatite:**

Associative  
Perusing of  
APIs  
That  
Identifies  
Targets  
Easily



**Crystal:**

Clarifications  
Regarding  
Your  
Software using a  
Toolkit,  
Architecture and  
Language



**Jadeite:**

Java  
API  
Documentation with  
Extra  
Information  
Tacked-on for  
Emphasis



**Jasper:**

Java  
Aid with  
Sets of  
Pertinent  
Elements for  
Recall



**Aquamarine:**

Allowing  
Quick  
Undoing of  
Any  
Marks  
And  
Repair  
Improving  
Novel  
Editing



**GARNET**

Generating an  
Amalgam of  
Real-time,  
Novel  
Editors and  
Toolkits



**Pebbles**

PDA's for  
Entry of  
Both  
Bytes and  
Locations from  
External  
Sources



**Gneiss:**

Gathering  
Novel  
End-user  
Internet  
Services using  
Spreadsheets



**Sugilite**

Smartphone  
Users  
Generating  
Intelligent  
Likeable  
Interfaces  
Through  
Examples



**Glacier**

Great  
Languages  
Allow  
Class  
Immutability  
Enforced  
Readily

**C32**

CMU's  
Clever and  
Compelling  
Contribution to  
Computer Science in  
CommonLisp which is  
Customizable and  
Characterized by a  
Complete  
Coverage of  
Code and  
Contains a  
Cornucopia of  
Creative  
Constructs, because it  
Can  
Create  
Complex,  
Correct  
Constraints that are  
Constructed  
Clearly and  
Communicated using  
Columns of  
Cells, that are  
Constantly  
Calculated so they  
Change  
Continuously, and  
Cancel  
Confusion

For more, see: [www.cs.cmu.edu/~bam/acronyms.html](http://www.cs.cmu.edu/~bam/acronyms.html)

# Thanks to:



Carnegie Mellon University



## • Funding:

- NSF under IIS-1116724, IIS-0329090, CCF-0811610, IIS-0757511 (Creative-IT), ITR CCR-0324770 as part of the EUSES Consortium
- SAP
- Adobe
- IBM
- Microsoft Research
- Yahoo! InMind
- Google



## • >38 students & visitors:

- |                   |                                  |                   |                                 |
|-------------------|----------------------------------|-------------------|---------------------------------|
| ■ Htet Htet Aung  | ■ Christian Dörner               | ■ Andy Ko         | ■ Sunyoung Park                 |
| ■ Jack Beaton     | ■ Dan Eisenberg                  | ■ Thomas LaToza   | ■ Michael Puskas                |
| ■ Ruben Carbonell | ■ Brian Ellis                    | ■ Joonhwan Lee    | ■ Chotirat (Ann) Ratanamahatana |
| ■ John R. Chang   | ■ Andrew Faulring                | ■ Toby Li         | ■ André L. Santos               |
| ■ Kerry S. Chang  | ■ Aristiwidya B. (Ika) Hardjanto | ■ Leah Miller     | ■ Christopher Scaffidi          |
| ■ Polo Chau       | ■ Erik Harpstead                 | ■ Steven Moore    | ■ Jeff Stylos                   |
| ■ Luis J. Cota    | ■ Amber Horvath                  | ■ Mathew Mooty    | ■ David A. Weitzman             |
| ■ Michael Coblenz | ■ Sae Young (Sophie) Jeong       | ■ Gregory Mueller | ■ Yingyu (Clare) Xie            |
|                   | ■ Mary Beth Kery                 | ■ Yoko Nakano     | ■ Zizhuang (Zizzy) Yang         |
|                   |                                  | ■ Stephen Oney    | ■ YoungSeok Yoon                |
|                   |                                  | ■ John Pane       |                                 |

# Thank You!

## Human-Centered Methods for Improving API Usability

---

**Brad A. Myers**

Human-Computer Interaction Institute

School of Computer Science

Carnegie Mellon University

<http://www.cs.cmu.edu/~bam>

[bam@cs.cmu.edu](mailto:bam@cs.cmu.edu)

