

# Agile GUI testing

*An Alexandrian solution?*

*Hans-J. Brede*

# Aims and obstacles

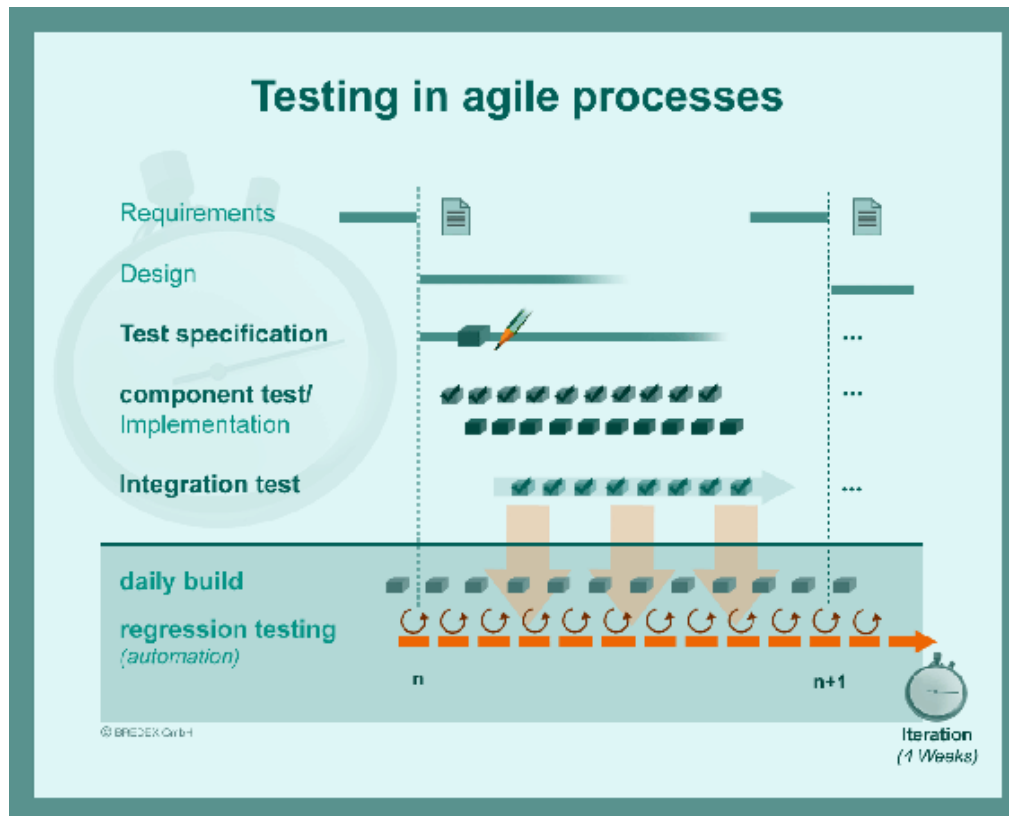
## ▶ Aims of software testing

Good quality software  
At a reasonable cost

## ▶ Obstacles to agile testing

Automation is necessary  
→ "I don't have time"

# The benefits of agile GUI testing



# More obstacles for agile GUI testing

## ► Frequent releases

There's not enough time to test!

I can't test what I can't see!

## ► Flexibility

Maintaining my tests is a nightmare!

Everything keeps changing!

# More obstacles for agile GUI testing

## ► Technical excellence, good design

Design and structure  
aren't important for  
tests!

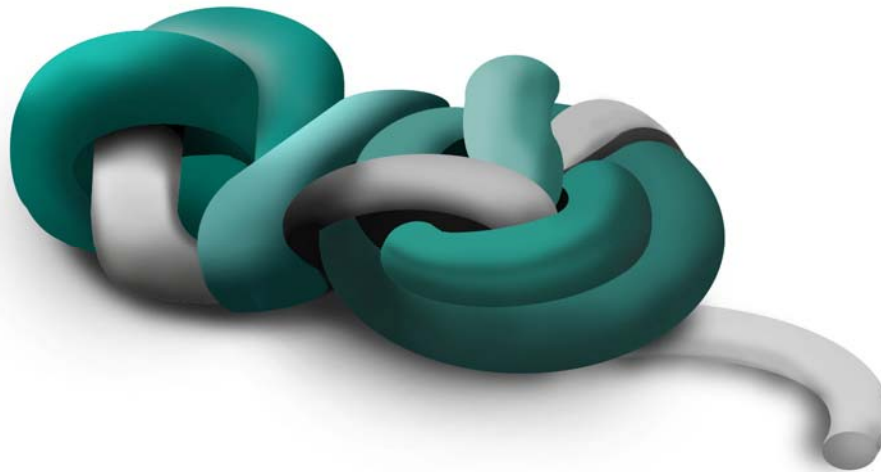
We need to start testing  
now, not waste time  
planning!

## ► Communication

The testers' job is  
separate from the  
rest of the team

We don't talk about  
strategy or details.

# Unsolvable problems?



# Agile GUI testing checklist

- ▶ **Start writing tests as soon as the requirements are available**

Unambiguous requirements documents

Communication

Start small – test what you know, use placeholders




Fill in the gaps as you go along



# Agile GUI testing checklist

## ► Structure your tests from the outset

Write modules to reuse

- ☐  Open Any Dialog From Menu [WHICH\_PATH, WHICH\_DIALOG]
  - ☐  <Select From Menu [WHICH\_PATH]>
  - ☐  <Wait For Dialog To Appear [WHICH\_DIALOG]>

Reuse them in more detailed modules

- ☐  Open Search Dialog From Menu
  - ☐  <Open Any Dialog From Menu [WHICH\_PATH, WHICH\_DIALOG]>



# Agile GUI testing checklist

## ▶ Structure your tests from the outset

Make use cases independent of each other

Use naming conventions

Refactor as you go

# Agile GUI testing checklist

## ▶ Plan for change

Keep test flow, test data and object maps separate

Think about data and when you may want to change it

→ What data varies case by case?

→ What data can be managed centrally?

# Agile GUI testing checklist

## ▶ Run your tests every night

Automated build process

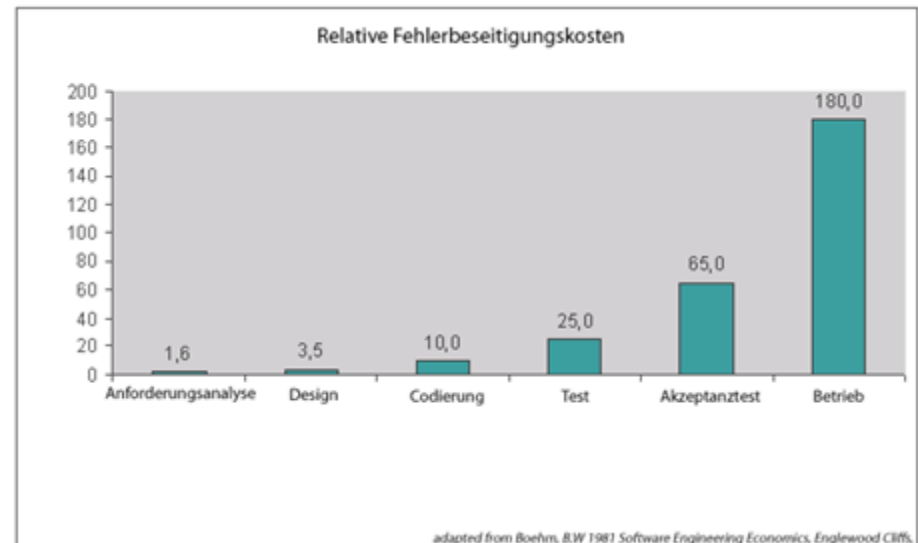
Environment

Test data, initial state of application

Error management

# Results of agile GUI testing

	P1 (agile)	P2 (non-agile)
Development (h)	1000	1000



# Results of agile GUI testing

	Assume: 25 errors in project, 0.5 days for error fix during implementation					
	P1 (agile)			P2 (non-agile)		
	#	h/error	Hours	#	h/error	Hours
Implementation	20	4	80	8	4	32
Test	4	26	104	16	26	416
After deployment	1	72	72	1	72	72
<b>TOTAL:</b>			<b>256</b>			<b>520</b>

## Summary

- ▶ **Testing agilely fulfills two aims:**
  - ▶ **1. Good quality software**  
Tested often and thoroughly from outset
  - ▶ **2. At a reasonable cost**  
Errors found earlier are cheaper to fix  
Well-structured tests are easier to maintain
- ▶ **What about the unsolvable problems?**

# A different approach is the key



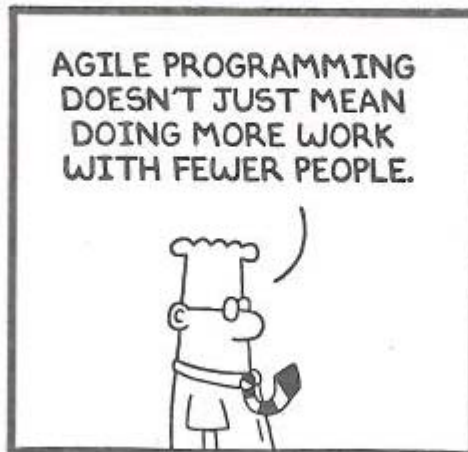


## An Alexandrian solution

- ▶ **Not having enough time is no excuse!**
- ▶ **Agile GUI testing is successful**
- ▶ **Changes to process**
  - Test at the right time*
  - Test in the right way*



**DILBERT**  
BY SCOTT ADAMS



1/14/05 © 2005 Scott Adams, Inc./Dist. by UFS, Inc.

