

HEXAGON SOFTWARE

Transition Testing

Cornerstone of Database Agility

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 1

HEXAGON SOFTWARE

Who I am
How this session runs

Setup

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 2

HEXAGON SOFTWARE Max Guernsey, III



- Eleven year veteran software developer/team leader
- Author *Transition Testing: Cornerstone of Database Agility*
- Creator *Database Agility Online Training* 

Nerd Eye for the HR Guy 

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 3

HEXAGON SOFTWARE Structure of Session

- Changed a lot since submission
- Three phases
 - Talk about principles
 - Demonstration
 - Q&A
- After the session
 - Slides, demo, and video all published to internet

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 4

HEXAGON SOFTWARE

JIT (Just in Time)


Agile Principles and Assumptions

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 5

HEXAGON SOFTWARE Principle: Deliver Fast & Complete


- Define requirements & specifications JIT
- Build only what is needed
- Refactor to support changes

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 6

 **Model: Evolution**


- Code and product improve over time
- New software installations replace old ones
- Kills out of date individuals in favor of the species

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 7

 **Practice: Compile, Test, Install**


- A common way to release binaries:
- Build binaries
- Test
- Deploy

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 8

 **"Deploy?"**

- New deployment?
 - Install
- Broken deployment?
 - Uninstall
 - Install
- Upgrade?
 - Uninstall
 - Install
- Creation-driven

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 9

 **Assumption: Easily Replaceable Product**


- Binaries : Products :: Individuals : Species
- We don't care about binaries
- ...but what if we did?

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 10

 **Mass**

New Forces in Data

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 11

 **Databases are not Easily Replaced**

- Certain databases are valuable
 - *e.g.*: Production, some test environments
- These databases cannot simply be replaced
- This violates the assumption made in most Agile activities

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 12

HEXAGON SOFTWARE Not Evolution: Metamorphosis

- Databases are *created* once
- Databases are *transformed* many times
- The evolutionary model fails us
 - Evolution is built on creative destruction
- The applicable model for change is metamorphosis
 - Allows entities to change shape while retaining identity

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 13

HEXAGON SOFTWARE One Way to Define a Database

- In production: a series of transformations are applied to a database
 - CREATE TABLE Foo
 - CREATE TABLE Bar
 - time passes
 - ALTER TABLE FOO
 - CREATE TABLE FUN
- Why not *start out* defining databases in these terms?
- The proper way to define a database is as a series of transformations
 - Nothing → v1
 - v1 → v2
 - v2 → v3
 - Etc
- Which reduces to...
 - → v1
 - → v2
 - → v3
 - Etc

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 14

HEXAGON SOFTWARE Building a Database out of Versions

- Proper way to build a DB:
 - Determine which transformations already applied
 - Apply remaining transitions in correct order
- Example: v2 → v5
 - → v3
 - → v4
 - → v5

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 15

HEXAGON SOFTWARE

Classes and Predictable Behavior

Repeatability

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 16

HEXAGON SOFTWARE TDD & Predictability

- There is no agility predictability
- There is no predictability without test coverage
- Might as well do it at the right time
- TDD is the foundation of repeatability

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 17

HEXAGON SOFTWARE Traditional Tests & Objects

```

classDiagram
    class MyTest {
        +TestFoo()
        +TestBar()
    }
    class MyType {
        +Foo()
        +Bar()
    }
    class consumer["consumer: Client"] {
        -UseObject()
    }
    MyTest --> MyType
    consumer --> MyType
    
```

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 18

HEXAGON SOFTWARE Testing Objects in Application Code

- With code (programs) testing a representative object is very meaningful
- The object under test is *prototypical*

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 19

HEXAGON SOFTWARE Confidence-Path for Classic Objects

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 20

HEXAGON SOFTWARE Traditional Tests & Data Stores

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 21

HEXAGON SOFTWARE Testing Database Instances

- With databases tests are not necessarily meaningful
- One database instance is not necessarily representative of others
- Tests are based on belief rather than reality


8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 22

HEXAGON SOFTWARE Confidence-Path for Most Databases

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 23


HEXAGON SOFTWARE Tests & Classes of Databases

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 24

 **Isn't a Series of Deltas Enough?**


- Necessary but not sufficient
- You also have to be confident in each change
- That's where transition tests come in

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 25

 **What are Transition Tests?**


- Transition tests are tests for shifts in database behavior
- Setup:
 - Build database up to a well-understood version
 - Populate with data
- Trigger:
 - Advance the database (usually) one version
- Assertion:
 - Prove that behavioral shift was completed
 - Prove that knowledge in database is still safe

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 26

 **Example Transition Test (Pseudo-Code)**

- Upgrade database to version **1.0**
- Populate column **X** in table **Y**
- Upgrade database to version **2.0**
- Column **X** in table **Y** must have moved to table **Z**


8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 27

 **Creating a Class of Databases**

Defining a database as a series of transitions
+ Testing each transition


A class of databases

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 28

 **Let's see it work**

Transition Testing in Action

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 29

 **A Moment about DataConstructor™**

- DataConstructor is a tool
- Define databases in terms of versions
- Use it to build databases
- It inspects databases to see what versions were applied
- Remaining transitions applied in correct order

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 30

 **Switch to Demonstration**

[Click here to switch to demonstration](#)

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 31

 **Where to Find Stuff**

- DataConstructor
 - <http://www.dataconstructor.com>
- Slides, Demo Zip, Video
 - <http://www.hexsw.com>

8/30/2009 © Copyright 2009 Harbinger Software Corporation. All rights reserved. 32