



Exploring, Discovering and Exterminating Bug Clusters in Web Applications

Kim Davis, MyVirtualModel, Inc.
kim@myvirtualmodel.com
www.myvirtualmodel.com

Robert Sabourin, AmiBug.Com, Inc.
rsabourin@amibug.com
www.amibug.com

Quality Week - 2001
San Francisco



AmiBug.Com, Inc.



Overview

- Method Motivation & Project Presentation
- Risks: Business, Technical
- Potential Areas of Instability & Test objectives
- Exploratory Testing & Mapping
- From Mapped Areas of Instability to Bug Clusters
- Finding, Prioritizing and Exterminating Clusters
- Conclusion
- Future work
- Web References

Motivation

- 80/20 Rule: Can we save lot's of time and effort by focusing on larger structures of bugs, i.e. *Bug Clusters*?
- Apply successive approximations
- Identify the cluster – without finding all of its individual bugs!
- Identify, Prioritize and Exterminate at the Bug Clusters Level
- Fix bugs that you did not find!

My Virtual Model Community

- **My Virtual Model Concept & Community**
- **3D, Web and Fashion Industries Collide**
- **Visualization**
- **Personalization – Virtual Identity**
- **Network of Interoperating Web Sites – Mobility**
- **Affiliate Services & E-Commerce**



My Virtual Model Community



 **My Virtual Model™**
Hello I'm QW2001, your Virtual Model.

 **My Virtual Model.com**

[MY HOME](#) [SHOPPING](#) [FASHION ADVICE](#) [HELP](#) [ABOUT US](#)



Fashionable clothing for **My Virtual Model™**

Click on the item of your choice to try it on your model, click again to remove the item.
You can change clothes by clicking a different article, without removing the previous one.



Can't get enough? Check our list of affiliated retailers for great [shopping destinations](#).



Business Risks

- Fixed timeframe projects tied to industry dates for seasonal collections of garments
- Community is large and demanding in terms of consistency, performance, reliability
- It has to work and it has to be on time
- Functionality always tied to market “\$\$” related revenues e.g. a new type of model can be introduced to target a new market (for example Tall Men)
- New business model is evolving as project continues!
- Organizational change
- Requirement turbulence



Example Project

- Constraint from moment clothing in-hand to release - max 8-12 weeks
- Community per client includes 100s of thousands of users - target large retail clients - we will get many many hits fast
- No forgiveness in retail fashion business - (nor Web business)!
- Trading down functionality to hit a release target is problematic
- Business requirement - release date moved up!


Technical Risks

- Technology change Management
 - Fast technology churn
 - Use technologies in production environments with limited knowledge of baseline
 - Limited time for evaluation, training
- Staff change
 - Programmer not familiar with entire code base - could miss something
 - Programmers not familiar with “live” production parameters - even if the developer knows how to make it work on his own machine (NT vs UNIX)
- Requirement turbulence

Example Technical Risks

- Many parallel activities not easy to synchronize
- New application server framework, business logic layer
 - From JServ to JRUN
- New presentation layer
 - to JSP model
- Some reused code at risk because running in a new context
- Changed requirements on technologies used

Testing Objectives & Risk

		Project Name:							
		Author:	KD						
		Revision:	3/30/01						
ID	Summary	Note	Proposed Owner	Relative Importance Suggested % of Total Testing Effort				Consequences of Failure (Point Form List)	
				Develop- ment Lead	Product Manage- ment	Testing Lead	Average	Business	Technical
Proj-HLTO-01	Functionality	Suitability, Accuracy, Interoperability, Compliance, Security	Product Management	30	20	25	24	Loss of confidence - High Maintenance & patches	High Maintenance
Proj-HLTO-02	Reliability	Maturity, Fault Tolerance, Recoverability	Product Management	20	30	30	27	Loss of confidence - High Maintenance	High Maintenance
Proj-HLTO-03	Efficiency	System behaviour over time, Resources Usage	Development Management	20	25	15	22	Loss of confidence - High Maintenance	High Maintenance
Proj-HLTO-04	Usability	Understandability, Learnability, Operability	Product Management	10	15	20	14	Loss of user interest - Loss of customers	
Proj-HLTO-05	Maintainability	Analysability, Changeability, Stability, Testability	Development Management	10	5	5	7	Maintenance Difficulty	
Proj-HLTO-06	Portability	Adaptability, Installability, Conformance, Replaceability	Development Management	10	5	5	7	Difficulty to get clients on different platform	Lack of interoperability
			Total	100	100	100			

Potential Areas of Instability

- Functions particularly related to new technologies
- New stuff or changed stuff (Turbulence Index)
- Functionalities added after design due to business change

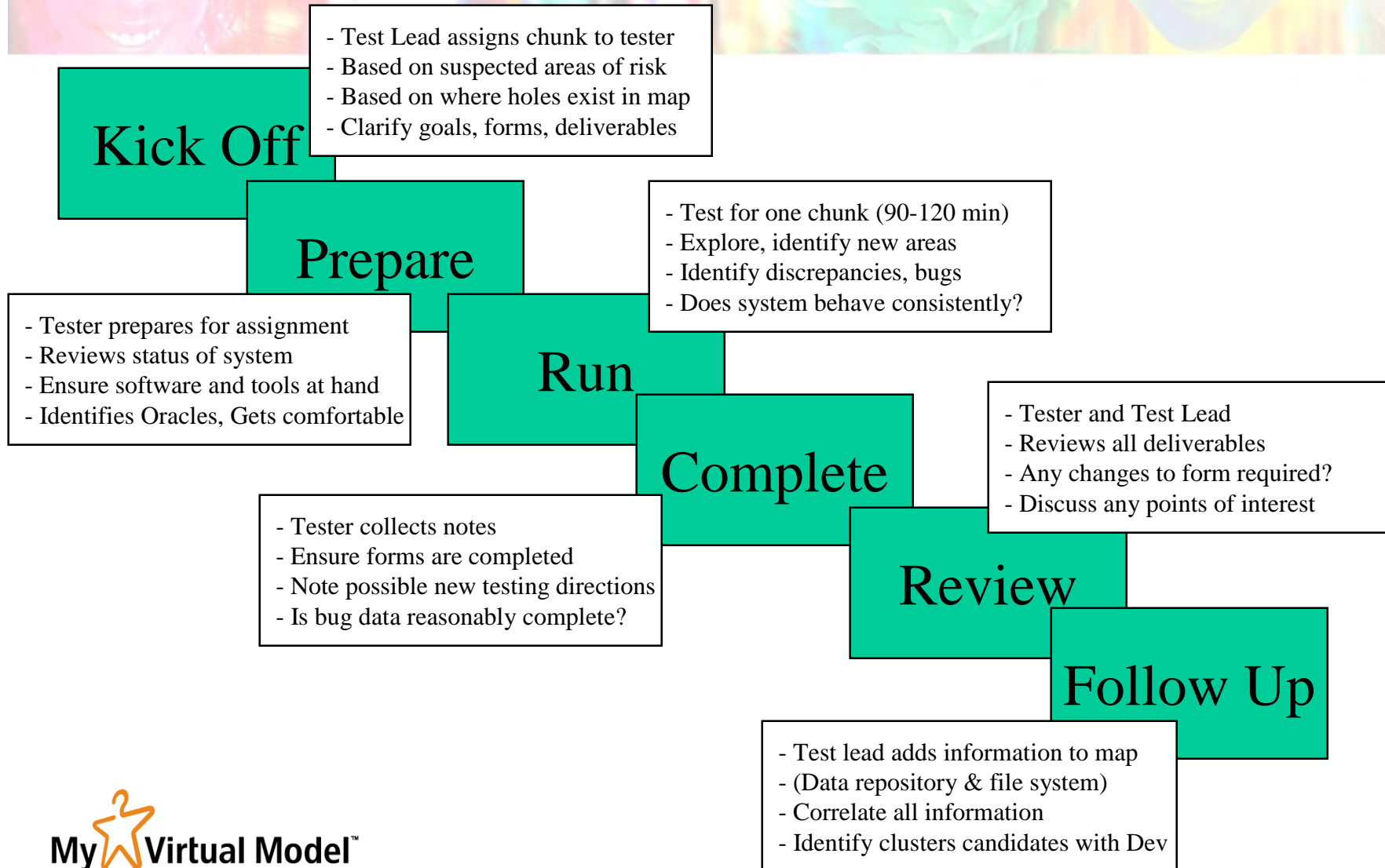
Exploratory Testing Approach

- *“In operational terms, exploratory testing is an interactive process of concurrent product exploration, test design and test execution.”*

- James Bach

- Test lead triage testing in chunks building up a “map”
- Assignment based on looking for areas of potential instability
 - Looking for clusters - observe aggregate results from all testers

Exploratory Testing



Finding a Cluster from Bugs Identified

- How should you/could you or do you identify a cluster?
- Correlation between bugs identified to get an understanding of whether they can be grouped
- Mapped Areas of Instability define clusters composed of the bugs that have natural affinities of causes
 - Requirement or Technology Turbulence
 - Copy-paste and reuse of code without checking

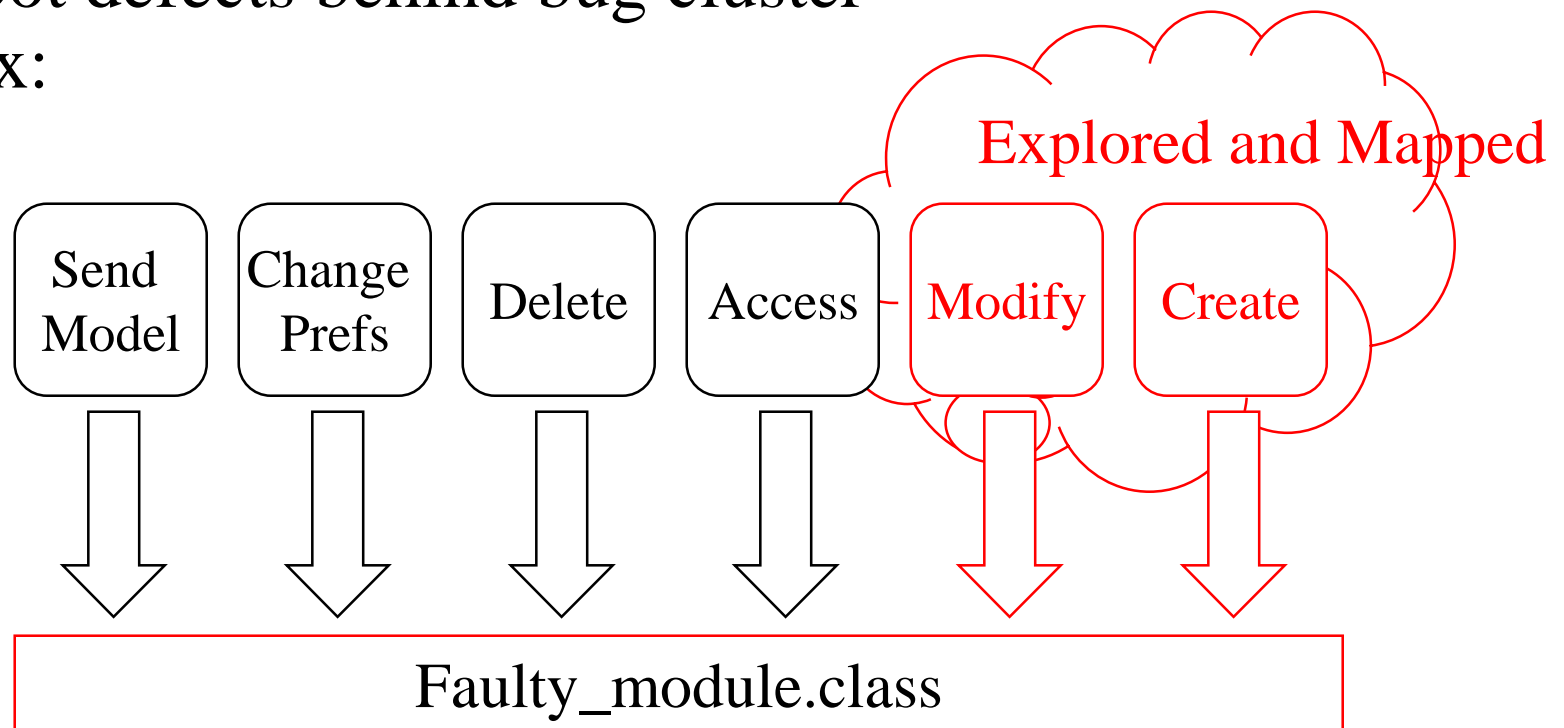
Clusters of Bugs

- What makes a Bug Cluster?
- Example Metrics Definitions
 - Functionality - several bugs related to same functionality are discovered
 - Reliability - different functions fail in similar way
 - Efficiency - several operations are similarly using resources inefficiently
 - Time - several content sources are out of sync
- Could indicate a process-related problem

Cause of Clusters

- Test team works with developers to find common root defects behind bug cluster

Ex:



Prioritize Based on Business Impact

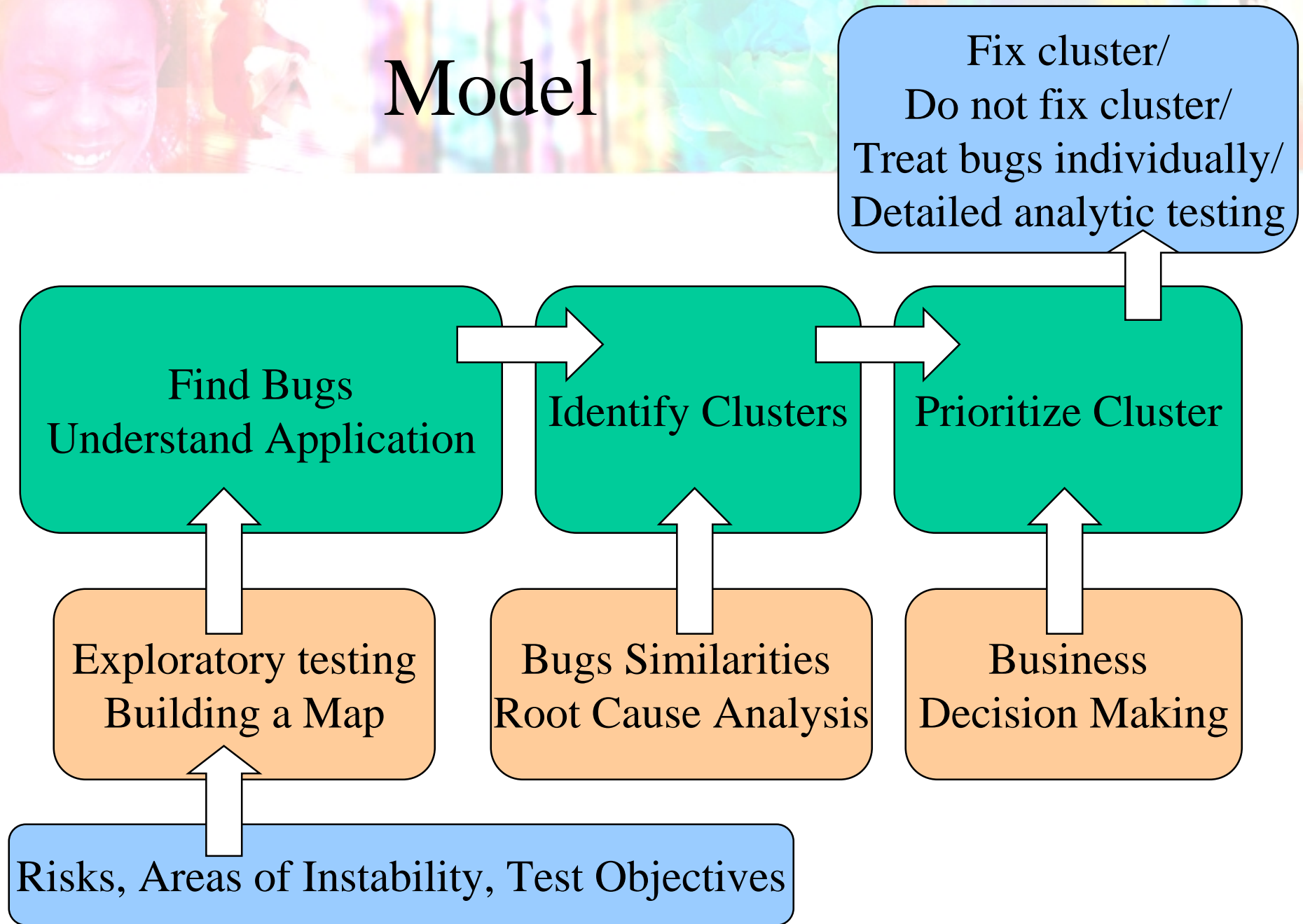
- Instead of prioritizing on a bug by bug basis, we work on a cluster (on a higher level)
- Less red tape
- Nice for speed!
- Can
 - Fix cluster
 - Do not fix cluster
 - Treat bugs individually
 - Detailed analytic testing



Practical Results

- Extermination techniques, developers and testers working together (fast)
- Try to identify probable root causes of sets of seemingly related bugs – Use empirical knowledge
 - Testers and developers worked well together
 - Improved developer awareness of tester role
 - Buy-in to approach
- Analysis based on root cause identified an additional 30%, so for every 10 bugs identified 13 were fixed
- **Continue using technique!**

Model





Conclusion

- Relevant in our context
- Shortening bug management overhead
- In less time we were able to deal with more bugs
- Flexible Method – Hybrid
- Buy in at all levels!
- In a turbulent environment, this will likely be a good approach on future projects



Future Work ...

- As method is applied to more projects we will better document and generalize it
- Exploratory Testing can be replaced by more analytical approaches as feeding method
- So far - developers, testers and management all like method - and the results
- Cluster Testing Method's Effectiveness & Efficiency – Using it to its fullest potential – Limits of applicability
- Will evolve based on REAL WORLD needs

Web Reference Slide

- **www.satisfice.com**
 - James Bach web site, exploratory and risk based testing
- **www.testing.com**
 - Brian Marick web site, articles about exploratory testing
- **www.amibug.com**
 - Robert Sabourin Web site, various presentations
- **www.mvm.com**
 - My Virtual Model Web site
- **www.stickyminds.com**
 - Much relevant content