

Additional Workshop Information

Workshop: Software Testing

Slides: The slides for this presentation were developed in Microsoft Powerpoint for Windows 95. They are available for downloading from the following web site:

<http://polaris.umuc.edu/~cfisher2/>

Follow the links to the STC conference presentations.

References: This presentation was compiled in part from information in the following sources. Use them if you would like more information on this topic.

Hetzel, Bill. *The Complete Guide to Software Testing*, 2nd Edition. New York: Jon Wiley & Sons, 1988.

Kaner, Cem. *Testing Computer Software*. Blue Ridge Summit, PA: TAB Professional and Reference Books, 1988.

Kit, Edward. *Software Testing in the Real World: Improving the Process*. Wokingham, England: Addison-Wesley, 1995.

Perry, William E. *Effective Methods for Software Testing*. New York: Jon Wiley & Sons, 1995.

Sommerville, Ian. *Software Engineering*, 5th Edition. Wokingham, England: Addison-Wesley, 1996.

Royer, Thomas C. *Software Testing Management: Life on the Critical Path*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

Contacts: If you have questions or need more information about this presentation, contact the presenters as follows:

Charles Fisher	Tracey Chiricosta	Tom Witherspoon
Datatel, Inc.	KBM Group	KBM Group
703-968-4603	301-587-7333	301-587-7333
cdf@datatel.com	tchiricosta@kbmgrou.com	twitherspoon@kbmgrou.com

Software Testing Terminology

Requirement - Condition or capability needed by users to solve a problem or achieve an objective.

Test data - Input data and file conditions associated with a particular test case.

Test procedure - Document defining the steps required to carry out part of a test plan.

Test validity - Degree to which a test accomplishes its specified goal.

Independent verification and validation (IV&V) - Verification and validation of software by an independent organization.

Inspection - A formal evaluation involving detailed examination of the software by a person or group other than the designer.

Validation - Evaluating software at the end of the development process to ensure compliance with requirements.

Verification - Evaluating software at the end of a phase to ensure that requirements established during the previous phase have been met.

Types of Testing

Unit testing - Testing of individual programs or modules as they are written.

Integration testing - Testing in which software elements are combined and tested until the entire system has been integrated.

Functional testing - Testing to verify that the functions of a system are present as specified.

Regression testing - Selective testing to verify that modifications have not caused adverse side effects.

Acceptance testing - Formal testing to decide whether or not to accept the software.

System Descriptions

System #1: Coastal Hunger Omission Management Program (CHOMP)

CHOMP is a PC-based marine animal feeding schedule and food inventory tracking system used by the Caribbean Bureau of Marine Life to track the supply of shark food, the feeding habits and health of individual sharks, and the shark feeding schedule at Marine World. The system runs on a networked PC system with a Windows interface.

The system is designed to automatically deduct the amount of food the sharks eat from the inventory of food and notify Marine World personnel when the inventory is low so they can restock. The system also keeps track of the amount of food each shark eats, and issues a warning if the amount of food a specific shark eats increases or decreases significantly. The system also schedules the daily feeding times and amounts of food so that personnel know when and how much food to give each shark. The system also tracks information about the general health and condition of the individual sharks. Marine World veterinarians can enter information about the medical conditions of each shark, current behavior characteristics, and other key health information.

The system produces statistical reports that provide Marine World management with information about the amount of food each shark consumes over time, trends of cost information, and the types of sharks currently kept at Marine World.

System #2: Best Education Enrollment and Registration System (BEERS)

BEERS is a Unix-based student enrollment management system used by Party State University to register students for courses. The software runs on a central Unix computer, with networked terminals distributed throughout the campus. The system uses a character-based interface with function keys for access to commands.

The system is designed to allow the registrar's office to enter information about upcoming courses: course number and title, number of credits, location and date/time, and the faculty member assigned to teach the course. Students can then register for these courses up to four weeks before each semester begins. The system generates course schedules for each student for each semester, and it also generates a roster of students for each faculty member. After the course is complete, faculty members can enter grades for students in their classes, and the system then generates grade reports for each student. The system also produces enrollment management reports that show registration and enrollment trends at the university. University officials use these reports to determine whether to expand or eliminate specific courses offered.

System #3: Apple Agenda!

Apple Agenda! is a MacIntosh-based add-on software system that works with the Mac version of Microsoft Word. With the Apple Agenda! add-on, users can easily create a simple or complex agenda for a short meeting or a multi-day conference.

Apple Agenda! lets users identify the topics they want to include in their meeting agenda and the time period to allow for each topic. They can also add information about the speaker for each topic and the meeting location. They can enter the date and start time for the meeting, then list the meeting topics. If users then re-arrange the topics, the starting and ending time for each topic is automatically calculated. If users schedule meeting topics for more than 2 hours without a break, the system suggests adding a break period into the agenda. Users can adjust this feature to be longer or shorter than 2 hours, or to disable it altogether. Other features of the system include the ability to track special equipment needed for the meeting, and a warning if a meeting location is double-booked. Reports include a list of speakers, a formal agenda for all attendees, a list of meeting locations, space utilization efficiency reports, and a list of equipment.

Exercise #1: Defining Test Objectives and Features to Test

Which system did you select to test?

- CHOMP
- BEERS
- Apple Agenda!

What type of testing do you want to perform on this system?

- Unit testing
- Integration testing
- Functional testing
- Regression testing
- Acceptance testing

What are the objectives of the test?

Which specific features of the system do you want to test?

Exercise #2: Developing Test Scenarios

System being tested:

- CHOMP
- BEERS
- Apple Agenda!

Feature	Test Scenario	Expected Result

Exercise #3: Reporting and Tracking Problems

To: Test Coordinator
From: Tester #1
Re: Testing CHOMP, version 1.0 beta
Date: May 1, 1998

I have completed unit testing of the Daily Feeding Log data entry screen. I found a new problem with the group of fields that lets you enter the amount of food each shark ate. If you enter the shark's name then press tab to go to the Amount of Food field, the cursor moves to the field for the next shark's name instead. The only way to get to the Amount of Food field is to use the mouse. We should change this interface problem to make it consistent with Windows standards.

To: Test Coordinator
From: Tester #2
Re: Retesting problem with BEERS 2.1
Date: May 2, 1998

I received the latest version of BEERS from the development team today, and that problem I found with the grades still hasn't been fixed. This is very serious and must be fixed before we deliver the product. I have told the development team about this again and again, but they don't seem to think it is a serious problem.

To: Test Coordinator
From: Tester #1
Re: Testing CHOMP, version 1.0 beta
Date: May 3, 1998

The problem I found a couple of days ago with the tab order on the Daily Feeding Log data entry screen has been fixed. We can go ahead and close this issue.

However, if we have time, it would be nice to have the Daily Feeding report sort by feeder rather than by shark. I know we didn't design it this way, but our beta testers have given us feedback that the report would be more useful if it came out this way.

To: Test Coordinator
From: Tester #3
Re: Problems with Apple Agenda!
Date: May 4, 1998

I have found the following problems with Apple Agenda! during testing:

- If you set the Break period to less than 1 hour on the Preferences screen, you do not receive a warning if you exceed that time period without scheduling a break. This does not seem to be a problem if your break period setting is more than 1 hour.
- When you try to print out the list of equipment for a meeting, the system locks up and the only option is to reboot. This is a serious problem, especially if you have not saved your agenda. I have not been able to consistently repeat this, but we should start looking into it now.

Sample Problem Report Form

Problem	
Description	
Screen(s)	
Status	<input type="checkbox"/> New <input type="checkbox"/> Open <input type="checkbox"/> Closed <input type="checkbox"/> Canceled <input type="checkbox"/> Returned to development <input type="checkbox"/> Fixed <input type="checkbox"/> Deferred
Type of Problem	<input type="checkbox"/> Design issue <input type="checkbox"/> Bug <input type="checkbox"/> Documentation <input type="checkbox"/> Hardware issue <input type="checkbox"/> Database issue <input type="checkbox"/> Enhancement
Severity	<input type="checkbox"/> Minor <input type="checkbox"/> Serious <input type="checkbox"/> Fatal
Priority (1-5, where 1 is highest)	
Version Number	
Assigned to	

Sample Problem Report Form

Problem	
Description	
Screen(s)	
Status	<input type="checkbox"/> New <input type="checkbox"/> Open <input type="checkbox"/> Closed <input type="checkbox"/> Canceled <input type="checkbox"/> Returned to development <input type="checkbox"/> Fixed <input type="checkbox"/> Deferred
Type of Problem	<input type="checkbox"/> Design issue <input type="checkbox"/> Bug <input type="checkbox"/> Documentation <input type="checkbox"/> Interface <input type="checkbox"/> Hardware issue <input type="checkbox"/> Database issue <input type="checkbox"/> Enhancement
Severity	<input type="checkbox"/> Minor <input type="checkbox"/> Serious <input type="checkbox"/> Fatal
Priority (1-5, where 1 is highest)	
Version Number	
Assigned to	

Sample Problem Report Form

Problem	
Description	
Screen(s)	
Status	<input type="checkbox"/> New <input type="checkbox"/> Open <input type="checkbox"/> Closed <input type="checkbox"/> Canceled <input type="checkbox"/> Returned to development <input type="checkbox"/> Fixed <input type="checkbox"/> Deferred
Type of Problem	<input type="checkbox"/> Design issue <input type="checkbox"/> Bug <input type="checkbox"/> Documentation <input type="checkbox"/> Interface <input type="checkbox"/> Hardware issue <input type="checkbox"/> Database issue <input type="checkbox"/> Enhancement
Severity	<input type="checkbox"/> Minor <input type="checkbox"/> Serious <input type="checkbox"/> Fatal
Priority (1-5, where 1 is highest)	
Version Number	
Assigned to	

Sample Problem Report Form

Problem	
Description	
Screen(s)	
Status	<input type="checkbox"/> New <input type="checkbox"/> Open <input type="checkbox"/> Closed <input type="checkbox"/> Canceled <input type="checkbox"/> Returned to development <input type="checkbox"/> Fixed <input type="checkbox"/> Deferred
Type of Problem	<input type="checkbox"/> Design issue <input type="checkbox"/> Bug <input type="checkbox"/> Documentation <input type="checkbox"/> Interface <input type="checkbox"/> Hardware issue <input type="checkbox"/> Database issue <input type="checkbox"/> Enhancement
Severity	<input type="checkbox"/> Minor <input type="checkbox"/> Serious <input type="checkbox"/> Fatal
Priority (1-5, where 1 is highest)	
Version Number	
Assigned to	