Been testing software for over 10 years
Started out as a Manual Tester
Moved to Automation testing
Now leading teams, defining quality in organizations.
Started as a reflection of how much software testing has changed over the last decade
Early on, applications were dependent upon the hardware they were created for. They were more designed than programmed. Software engineering first appeared in the late 1950s.
The “software crisis” arose in the 1960s from poor performing applications.

- Hardware costs started to fall  
- but the cost of software development began to rise.  
- Many software projects  
  did not scale  
  were over budget,  
  late  
  or just plain failed.

- There was a need for quality software
The modern age of software testing began in the early 1970s. Dr. Winston Royce wrote, "Managing the Development of Large Software Systems". The foundation of the waterfall model was based on this paper.

At first, almost all software projects used the waterfall model. Phases had independent estimates and could not start until the application passed the previous phase.
- In the paper, “The Growth of Software Testing” classified software testing into different time periods.
- Each period had different goals and methods to define software quality.
- Modern testing was built on a combination of these periods.
- The first software was dependent on the underlying hardware.
- The terms debugging and testing were interchangeable.

**The First "Computer Bug"** Moth trapped between Relays, of the Mark II Aiken Relay Calculator while it was being tested at Harvard University, 9 September 1947.
- 1957 was the first year that testing was distinguished from debugging by Charles Baker.
- Testing during this period was focused on problem-solving and demonstrating that the application behaved as expected.
In his 1979 book, *The Art of Software Testing*, Glenford Myers defined testing as “...the process of executing an application with the intent in finding errors.”

In this period, the primary intent of testing was to cause fault in the application.
In 1983, the National Bureau of Standards published guidelines for federal information processing systems. The process integrated analysis, review and testing to evaluate the application.
- In 1988 a new methodology was written by the Institute of Electrical and Electronics Engineers (IEEE) unit
- “Systematic Test and Evaluation Process” or STEP.
- included test activities such as test planning, analysis, test design, and test plans with the goal of preventing defects.
Starting in the 1990s there were a few iterative methodologies proposed, such as eXtreme Programming (XP) and Scrum.

In 2001, we had the Agile Manifesto, which provided a method for a lightweight software development cycle to deliver software quicker than waterfall. Since then, multiple methodologies are now grouped under the agile method, including Scrum, Kanban, XP, and Lean.

These agile methods allow shorter delivery times and the ability to adjust to changing business demands.
User Stories

Requirements used to be large documents
User stores are small isolated bits of functionality
Often have done when or acceptance criteria
Cross-Functional Teams

Teams are now cross functional
Working together with the business owner, developers, QA
Most individual testing methods, only a 35% effective rate at catching defects

If testing is the sole preventative measure for a large scale project, it will seldom top 80% of defect removal.

A combination of quality methods, including pre-test methods such as static analysis for defect detection, can bring defect removal rates up to 95%.
How do we determine quality across different types of testing
Determine Quality

How Google Test Software, Dr James Whitaker
Determine Quality

ACC Model

- Attributes
  - Fast, Secure
- Components
  - Accounts, Search
- Capabilities
  - Login, Logout

This list of capabilities can be tied to individual tests
Automation/Unit Tests/Manual Testing
Manual Testing

- Test Cases
- Exploratory Testing
- Scenario Based
  - Test Tours
Exploding Configurations

Browsers

- Over the years exploding configurations
- A 2 hour acceptance tests on a configuration as example
- Going to use Browsers as example
Exploding Configurations

Browsers 2000

IE6 85%
Netscape 15% and declining

2 Configurations
Exploding Configurations

Browsers
2008

iPhone and Chrome Introduced
6 Configurations
Exploding Configurations

Browsers
2010

Android Introduced
8 Configurations
Exploding Configurations

Browsers
2011

9 Configurations
Exploding Configurations

What most people deal with
12 Configurations
If you are lucky and based your testing off browsers with greater than about 5%
8 Configurations
Exploding Configurations

- Operating Systems
- Languages/Locales
- Hardware/Mobile Platforms
- Resolutions

Other causes of exploding configurations
Know where testing takes place
Lisa Crispin Agile Testing
Maintainable
Reusable
Reliable
Right tool for the job
Quality at All Times

Continuous Integration

Jenkins

Atlassian
Bamboo

TC
TeamCity
Common Mistakes

- Agile is not mini waterfall
- Lack of (any) documentation
- Loss of traceability or coverage
- Quality ownership
  Who is responsible for quality?
  Everybody.

Documentation is communication

Quality is owned by the cross functional teams.
Next Generation
Skills in Demand
If you ever work in a start-up or with cutting edge technology, you will most likely be working in Open Source.
The hottest automation technologies (Cucumber, Watir, Selenium, Robot Framework, Sikuli) are all Open Source projects.
Scripting is often not enough. 
Programed in an Object Oriented language like Java, Ruby or Python
Skills in Demand

- Unit Test
- Continuous Integration
- Mobile
- Social Media
Next Generation

The Testers

Hybrid Testers
Leverage automation for repetitive tasks
Focus on exploratory testing, functional testing, and user experience
Next Generation

The Tools

Automation will be easier to maintain (frameworks)
Improvements in tools
Self Testing Webpages
Next Generation

The leadership

Centers of Excellence
Quality Architects
A lot has changed over the last 10 years
Testing methods that worked then may not work well anymore
Tend to hold on to legacy tools, like Quality Center

This is a primer, all the references in my paper.