Game Developers Conference® | **March 9-13, 2010** | Moscone Center | San Francisco, CA

THINK SERVICES
A DIVISION OF UNITED BUSINESS MEDIA LLC

GDC 10

Learn. Network. Inspire.

www.GDConf.com

QA’s 10 Commandments: What?! Only 10?

- A review of QA’s best practices and an examination of potential additions.

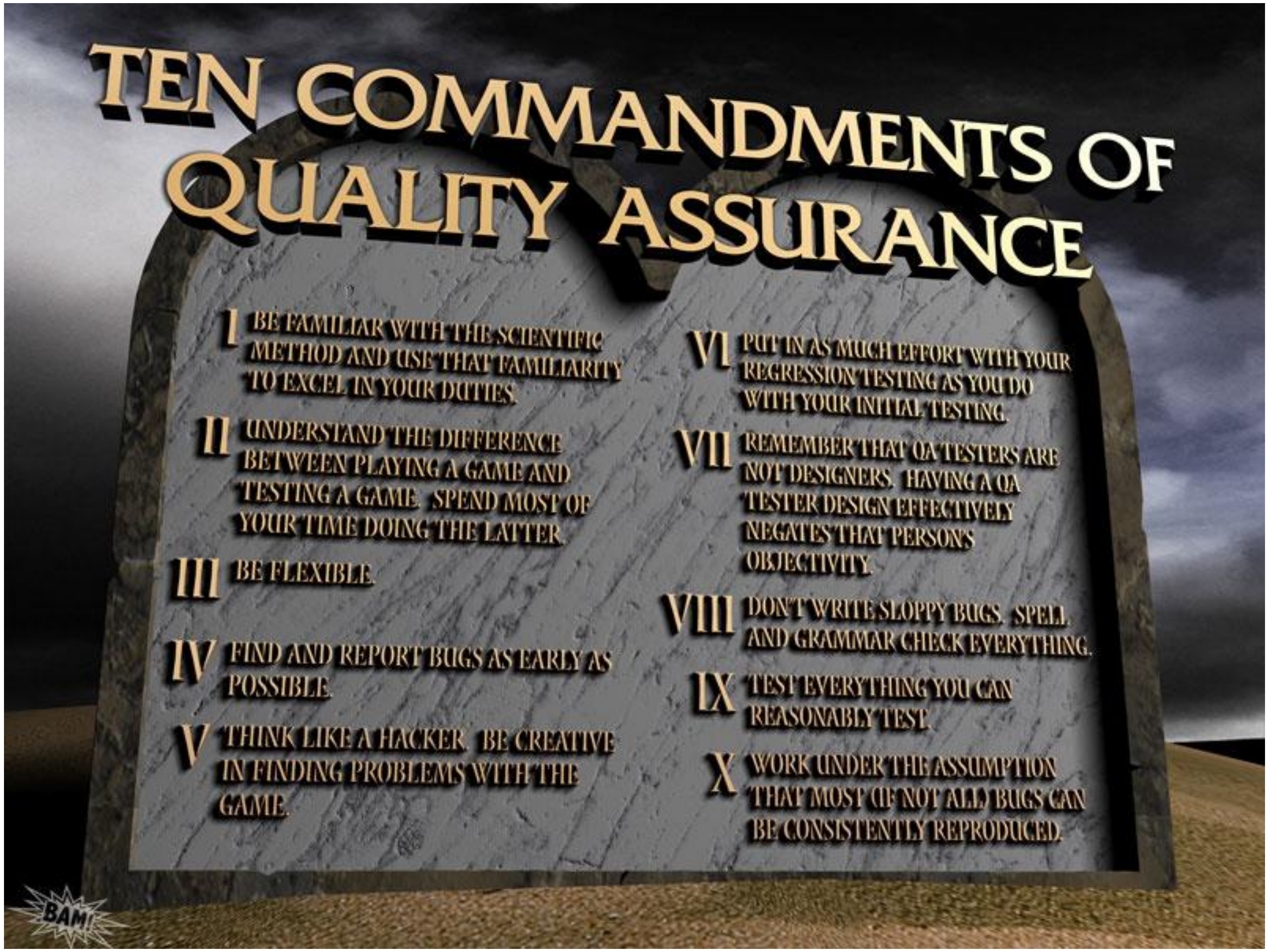
- Chuck McFadden

  Sony Computer Entertainment America

This lecture is NOT:

- An in-depth examination of **SCEA**'s QA practices or of PlayStation Home.
- A review of console submission requirements.
- An hour long.

The origin of the 10.



TEN COMMANDMENTS OF QUALITY ASSURANCE

I BE FAMILIAR WITH THE SCIENTIFIC METHOD AND USE THAT FAMILIARITY TO EXCEL IN YOUR DUTIES.

II UNDERSTAND THE DIFFERENCE BETWEEN PLAYING A GAME AND TESTING A GAME. SPEND MOST OF YOUR TIME DOING THE LATTER.

III BE FLEXIBLE.

IV FIND AND REPORT BUGS AS EARLY AS POSSIBLE.

V THINK LIKE A HACKER. BE CREATIVE IN FINDING PROBLEMS WITH THE GAME.

VI PUT IN AS MUCH EFFORT WITH YOUR REGRESSION TESTING AS YOU DO WITH YOUR INITIAL TESTING.

VII REMEMBER THAT QA TESTERS ARE NOT DESIGNERS. HAVING A QA TESTER DESIGN EFFECTIVELY NEGATES THAT PERSON'S OBJECTIVITY.

VIII DON'T WRITE SLOPPY BUGS. SPELL AND GRAMMAR CHECK EVERYTHING.

IX TEST EVERYTHING YOU CAN REASONABLY TEST.

X WORK UNDER THE ASSUMPTION THAT MOST (IF NOT ALL) BUGS CAN BE CONSISTENTLY REPRODUCED.

Use the Scientific Method.

- Observe and describe.
- Formulate a hypothesis.
- Experiment.
- Draw a conclusion.

- *This Commandment separates the good testers from the bad.*

Playing vs.Testing.

- Know the difference! Spend most of your time testing.
- Check your ego at the door: Test “losing” conditions as much as the “win” conditions.

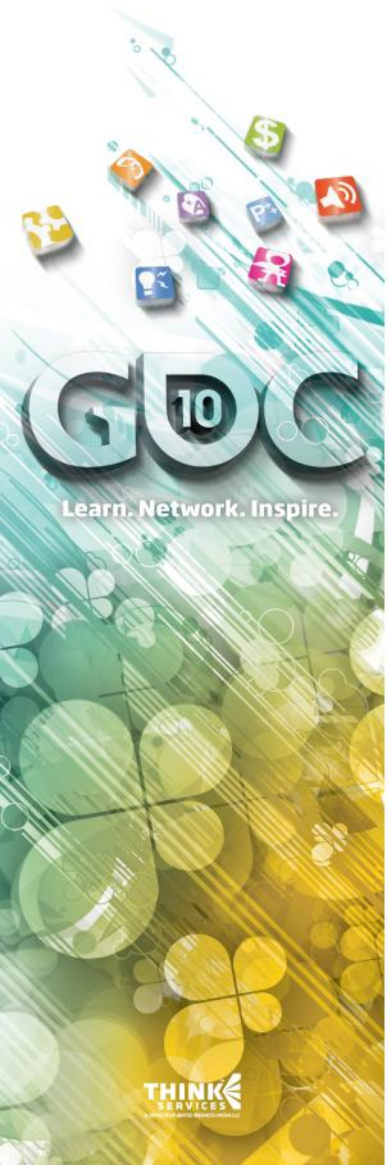
Be Flexible.

- QA testers/teams are used for a variety of tasks outside of traditional "testing." Encourage this.
- As a game nears completion, continually evaluate QA's bugs and reprioritize as necessary.

- *Recommendation: Use a prioritization scheme.*

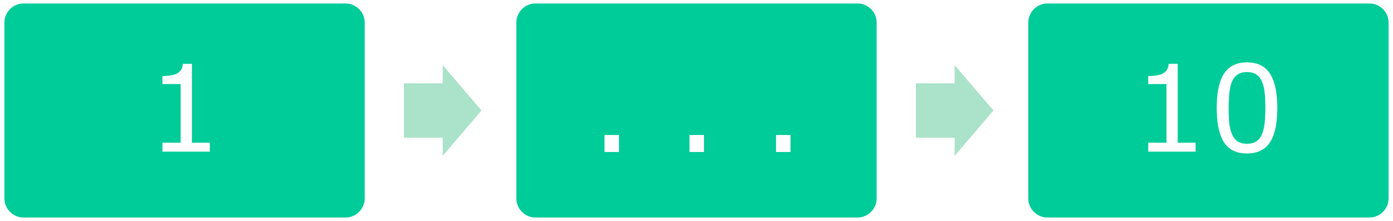
Find and Report Bugs as Early as Possible.

- Review the “save flow” when it’s a simple design on paper.
- Look at early UI text to spot incorrect usage of platform naming conventions.
- See if a tester can “find the fun” in an early build/prototype.



Think like a Hacker.

- Be creative in finding problems. Look beyond the surface.
  - Examples: Exiting a room through a door, window, ceiling, or wall.
- Don’t simply test how the game is “supposed” to be played.



Think like a Hacker (cont.)

- 1-10 might be tested thusly:
  - Test the #1, then the #2, then #3, and so on until 10.
- What happens when you test #0?
  - Or #-1?
  - Or #1,002?
  - Or #0.356?

Put in as much effort with your Regression as with the initial tests.

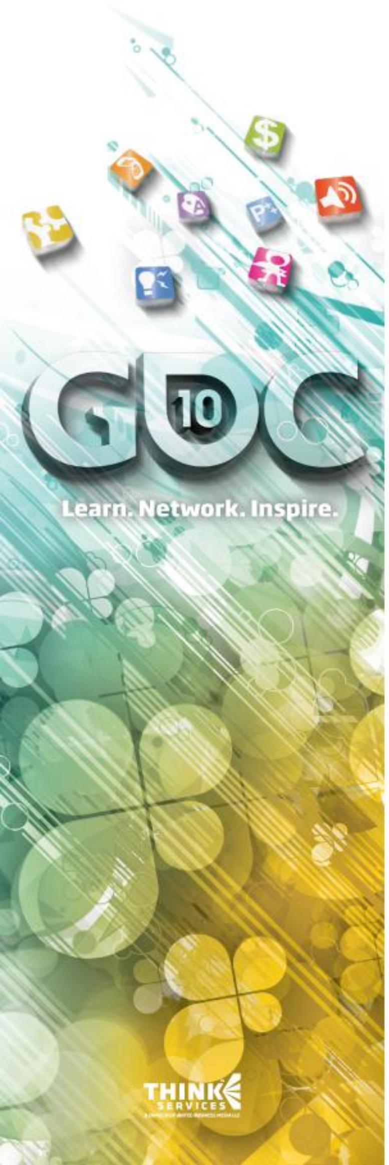
- Also known as "Halo testing."
- Check for new bugs that are a result of the fixed bug.
- Risk is *always* associated with any bug fix.

Don’t let QA members test designs they’ve championed.

- QA testers are only human.
- When a tester’s design input becomes an in-game reality, he/she *cannot* be allowed to test it.

Don’t write sloppy bugs!

- What is a tester’s most valuable weapon?
  - Awesum gamr skillz?
  - Effective communication?
- Spell and grammar check everything.
- Ensure “steps to repeat” are clearly and concisely written.
  - Any developer who follows the steps should be able to repeat the bug.



Test everything.*

- Never assume any feature is bug-free.
- Use test plans to help you test every feature in every (reasonable) way.
- Good enough isn't.

- ** Perhaps not such an important rule any longer. More later.*

Assume all bugs can be consistently reproduced.

- It's not a question of whether it's reproducible, rather an ROI question.
- Scientific Method helps a lot here.
- If the tester doesn't have enough time to consistently reproduce, make sure the bug details a percentage.

But, wait! There's more!

- Possible additions:
  - Automate tests.
  - User testing.
  - The End is the Beginning.

  * Test Everything.

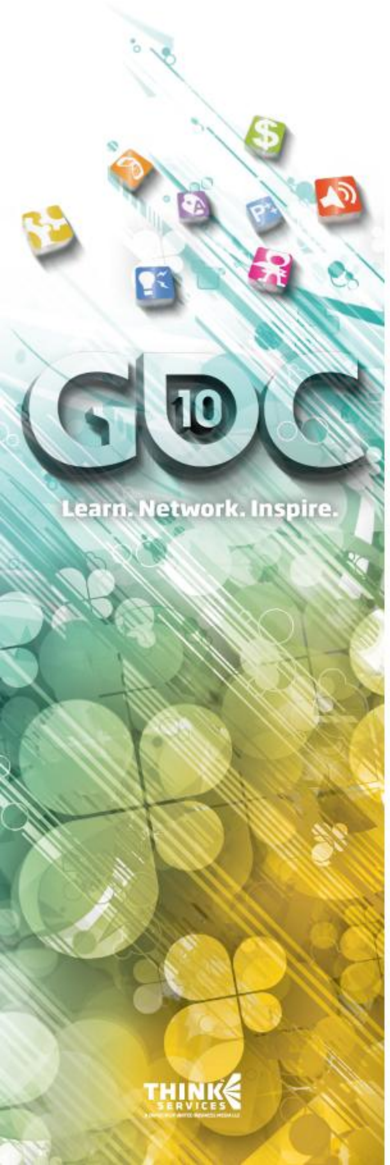
Automate Tests

- Identify tedious QA tasks that can be handled automatically.
  - Hire a QA tools programmer.
  - Encourage the dev team to devote some time to the task.

- *Key point: Automated tests are only helpful if they're planned for in advance.*

User Testing

- It’s not QA testing!
  - But it’s equally important.
- Don’t sour the user tests with your preconceptions.
- Learn to interpret “usability” feedback.
  - What users want isn’t always what they need.



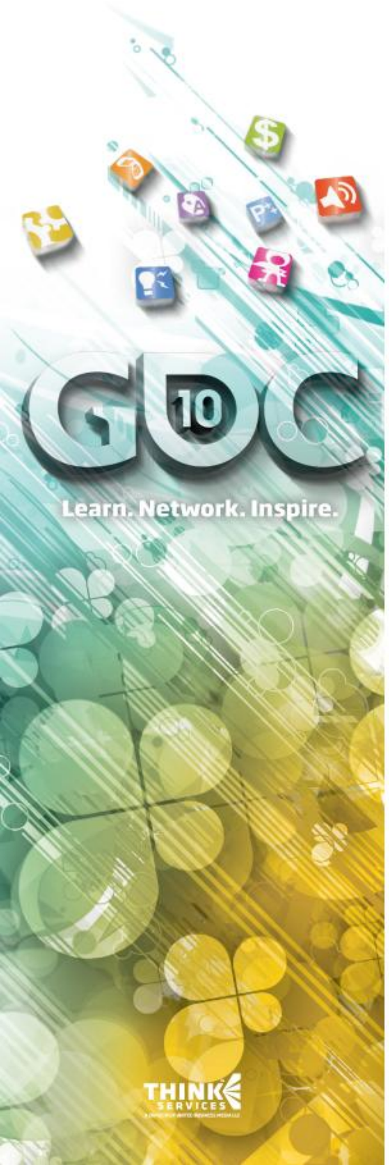
The End is the Beginning.

- Increasingly, games see post-release updates.
- Should your test strategies change?
  - Leverage modern business practices and technology to streamline your work.

*Test Everything.

- Food for thought:
  - Now that post-release development is common, is this commandment necessary?
  - Is “good enough,” good enough?

Game Developers Conference®
March 9-13, 2010
Moscone Center
San Francisco, CA
www.GDConf.com



Thank You!

- Questions?
- Comments?
- Performance Art?