



# Context-Driven Testing

James Bach  
Satisfice

<http://www.satisfice.com>

Michael Bolton  
DevelopSense

<http://developsense.com>

TestNet  
January 2012

# Cultures of Software Testing



- Factory (a.k.a. “Routine”)
  - minimize reliance on skilled people; follow templated processes; depend on documents
- Mathematical (a.k.a. “Analytical”)
  - intensive and rigorous mathematics, functional analysis and graph theory will guide us best
- Quality Assurance (a.k.a. “Process”)
  - requiring other people to adhere to rigid processes and driving them to high standards is the real goal; testing is incidental
- Agile
  - rely on skilled programmers; no “specialists” (that is, no non-programmers)



## What Does “Context-Driven” Mean?

“Context-driven testing” means that to be effective, testing must consider context *first*.

## Context-Driven Testing History

(all dates approximate)



- 1983: *Testing Computer Software* drafted
- 1988: *Testing Computer Software* published
- 1995: Kaner and Bach meet
- 1997: Los Altos Workshops on Software Testing begin
- 1999: Context-Driven School declared
- 1999: Context-driven testing mailing list set up
- 2002: *Lessons Learned in Software Testing* published
- 2003: “Four Schools of Software Testing” presented at the Workshop on Teaching Software Testing
- 2005: Black Box Software Testing online course launched
- 2011: The Conference of the Association for Software Testing focuses on context-driven testing

## Principles

1. The value of any practice depends on its context.
2. There are good practices in context, but there are no best practices.
3. People, working together, are the most important part of any project's context.
4. Projects unfold over time in ways that are often not predictable.
5. The product is a solution. If the problem isn't solved, the product doesn't work.
6. Good software testing is a challenging intellectual process.
7. Only through judgment and skill, exercised cooperatively throughout the entire project, are we able to do the right things at the right times to effectively test our products.

## People

- People, working together, are the most important part of any project's context.
- Good software testing is a challenging intellectual process.



All work of any value is done by and for people.  
At best, processes and tools *assist* that work.

## Practices

- The value of any practice depends on its context.
- There are good practices in context, but there are no best practices.



Would we test these things in the same way?

## Projects

- Projects unfold in ways which are often unpredictable.



Despite our best plans and intentions, there are sometimes surprises along the way, and contexts change.

## Products

- The product is a solution. If the problem isn't solved, the product doesn't work.



Yes, this is a joke. But think about it.

Similarly, if testing fails to provide value to clients, then testing doesn't work either.

## Tying It All Together

- Only through judgment and skill, exercised cooperatively throughout the entire project, are we able to do the right things at the right times to effectively test our products.





When might it NOT be a good idea to use context-driven testing?

**When might it be a good idea NOT to be context-driven?**



- When someone else is entirely responsible for the quality of your work.



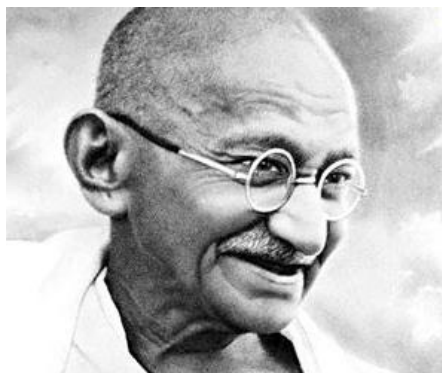
## When might it be a good idea NOT to be context-driven?

- When you're working in a single, specific, well-established and unchanging context.



## When might it be a good idea NOT to be context-driven?

- When your goal is to *change* the context.





## Implications of the Principles

Testing groups provide testing-related services.  
They do not run the development project;  
they serve the project.

## Testers Light The Way



**This is our role.**

*We see things for what they are.  
We make informed decisions about quality possible,  
because we think critically about software.*