

STORIES FOR REGRESSION TESTING

6 IDEAS

REGRESSION TEST LIKE A GARDEN

- DESIGN
- BUILD
- WEED
- REDESIGN
- LIMITING OF -SITING
-ALL

LAYER & SELECT

SESSIONS OF EXPLORATORY TESTING

REMEMBER THE WHOLE (EZE) - DON'T JUST REG. TEST THE BIT

(USE REGRESSION TEST TO LEARN THE SYSTEM (S))

DON'T FORGET THE ASSERTIONS

- BE CAREFUL OF BAD AUTOMATION

"GREEN IS NOT
NEARLY ENOUGH"

LEARN FROM YOUR USERS / CUSTOMERS

DOWN

- Not being good enough
 - safe place to learn / fail
 - ask for help
 - practice
 - no finger pointing
- Demotivation
 - is it, how can I get myself ^{reinvigorated}
 - colleague: how can I help him.
- Things outside my control / being asked to do too much
 - Learn to say no / not yet.
 - Get ~~better~~ ^{better} at influencing.
 - Pull versus push

INTERESTED TESTING

FREE / EXPLORATORY ✓ ✓

GET THE TEAM INVOLVED
(NOT ONLY TESTERS)

ASKING QUESTIONS

CHALLENGING ✓

NOT REPETITIVE

PAIRING

Digging Deep to get to the bottom of the bug!

Getting to use new tools / stuff.

Learning about new things / apps.

CODE REVIEW

Curious

Looking Outside the box

Learning
Interacting with people
GO OUT / CHANGE THE PLACE"
BUILD A GOOD TEAM SPIRIT

No continuous testing - Play TableTennis
to Refresh

Playing games/competing
Take on a different personality

Pair Testing
CELEBRATE SMALL VICTORIES ^{More!} ^{Fun!} ^{DELL!}
ANY THING THAT MOVES FORWARD

SIREN ON RED BUILD.

USE NEW TECHNIQUE.

BUILD GOOD RELATIONSHIP W/ COLEERS
GLINT

KNOWLEDGE SHARING

WORK WITH CUSTOMER

FIND RIGHT TOOL: AUTOWRITE THE
DOLL

- Showing new functionality to users
- Automate the boring stuff
- Finding bugs in Automation.
(and fixing)
- Optimising {Automation
 {Any approach/technique}
 → to fit the context better)
 my hood.
- Selling my ideas
- Training / Mentoring people.
- Estrating with developers
 - get to under perspective
- Not having emergencies (in production)
 Having emergencies. ↓
 positive
- Conferences / meetup / community of
 practices

TEST OLD IN A NEW WAY.

BE CREATIVE

IMAGINE THE WORST CASE
GO TO EXTREME

TEST BEFORE IMPLEMENTATION
- ASK QUESTIONS

BRAIN STORMING + PLANNING + SHARING

AVAILABLE INFO

■ SPECIFYING TEST, COMING UP WITH
LOGICAL SITUATIONS THE BUSINESS
DID NOT THOUGHT OF

TESTING CLINIC

Empowerment + Responsibility

Principle:
+ Specific
+ Measurable

- REFUSE TO SIGN OFF
- TAKE ACCOUNTABILITY
- MAKE SURE OTHER ^{HAVE} ENOUGH INFO TO MAKE THE RIGHT DECISIONS.
- SHARE

• Getting profit (emotional / real) from
the business for a job well done,

• When we make someone else happy.

Getting free coffee.

INTERESTED TESTING

Plenty side tracks

+1 Finding & identifying Root cause of tricky bugs

+1

More Exploratory

FEEL MORE CLEVER!

GENERATE MANY TEST CASES

GETTING THE FEELING THAT WE MOVE FORWARD
(SAME BUGS DO NOT POP UP OVER & OVER)

Sapient testing

Having fun & following curiosity

+1 Feel challenged to use my
brain

CRITICABILITY OF TESTS

CHALLENGES & COMPLEXITY TO OVERCOME
WITH CREATIVE SCENARIOS

FUN

- Throwing things (without / with ^{writing} people)
- Exploring
- Debating / Discussion / Arguing.
- Working in groups with different people
- Learning
- Breaking stuff.
- Making jokes of stupid ^{software} people.
- Sub-text in bug reports / Easter eggs
- Playing games
- Switching roles
- Solving problems

EMPATHY + ROLE-PLAY

STORIES (REAL)
INDEX (TO TRY)

Hand puppet

Listening allows
people to feel
heard

Changing roles

↳ FASTER
ORGANISATION

Retro's

Be interested

Crosscultural

Spontaneous

Communication

Feel the mood

Get to know
each other

= CULTURE

DULL TESTING

No Bugs found - only repetitive testing & expected result

STICKING TO PLAN - STRUCTURED

EASILY IRRITATED

Missing details

MANUAL REGRESSION - 1 checklists

NON-CREATIVE

NOT READY TO TEST
(TOO MANY BUGS)

NO URGENCY (NON-CRITICAL TESTS)

NO PURPOSE

VERY LONG SCRIPTS

Humans acting like machines

- Trying to get information from Developers.
- " " meeting with developers
- Getting to the one version of the truth
- Getting involved in making the decisions
- Saying "I told you so"
- Proving it doesn't work on your machine
- Motivating developers to test (at all)
- Finding the unknown unknown
- Finding big bugs.
- Saying it is good to go.
- Getting a thank you from a developer

WHAT CAN WE CHANGE?

BIG LIST

- FIND WAYS TO HAVE MORE FUN
- PERCEPTION OF TESTING
- USE RETROSPECTIVES TO GET BETTER AT FINDING MORE BUGS
- USE AGILE PRACTICES IN THE INDEPENDENT TEST TEAM
- USE TRAINING INDICATIONS IN EARLY ^{ALTO} FUNCTIONAL TESTS TO GET PERFORMANCE INFORMATION
- USE COMMON CLOUD BASED TEST TOOL - SERVICE
- USE MORE REALISTIC ENVIRONMENTS & DATA

- Trying to get information from Developers.
- " " meeting with developers
- Getting to the one version of the truth
- Getting involved in making the decisions
- Saying "I told you so"
- Proving it doesn't work on your machine
- Motivating developers to test (at all)
- Finding the unknown unknown
- Finding big bugs.
- Saying it is good to go.
- Getting a thank you from a developer

DULL TESTING

- Regression testing manual
 - ↳ MAKE IT AUTOMATED (myself)! ↗
- setting up test data. ↗
- Test without thinking (dull steps to follow)
- Not having time (as a test manager) to test things myself!
- Test bad mobilephone versions/browsers
- Testing legacy systems
- Run test cases - [Monkey TCs]
- Testing reports
- Testing good software (no bugs!!)
- Low Risk & high likelihood to just work as expected...

TOOLS TO CRUNCH TEST TASKS

10 EXAMPLES
TOOLS

CA Agile REQUIREMENTS DESIGNER

CA TEST DATA MANAGER

HP ALM

JIRA / HANSOFT

HP UFT CUCKOO (SECU)

ROBOT / RIDE EYES + BRAINS (COFFEE)

SELENIUM / CUCUMBER / GHERKIN

JENKINS
NOTEPAD ++ / EXCEL
PAIRWISER

PEN & PAPER

APP IUM

LOADRUNNER
JMETER / BLAZEMETER
SMARTBEAR LOADUI

Getting engagement from developers

Telling users they are getting it wrong

Using new approaches / techniques

Investigating new tools

• Helping someone learn how to test

• Getting developers to do the testing

• Learning from other perspectives

Getting to know your users

Getting to know the market for similar products

• Challenging myself.

• Helping people to understand quality assurance and it's not just a tester's job

BUILD MUTUAL RESPECT
+
TRUST

MAKE PEOPLE HAPPY WITHOUT

BRING GOERS TO TEST

BRAINSTORM U/CODERS

SHARED FOCUS ON EXCELLENCE

SHARE END PRODUCT

↳ HOLISTIC

↳ HAPPY FLOW SPAN SYSTEM

THEMED SHARED TEAM OVERTIME
+ MGT COOK!

IN ROW, SILENT PROGRESS
+ FOCUS ON JOBS

COMPLIMENT ~~TEAM~~ COLLEAGUES

BREAK IT!

DO WHAT YOU WANT

CRUNCH TIME

TEST NEW THINGS.

TEACH IT

UP

→ * GIVE RECOGNITION & FEEDBACK

→ * COLLABORATION &

PROACTIVE TEAMWORK

* * LEARNING & BEING CURIOUS

→ * OPENNESS TO CREATIVITY & LEARNING

COMPETITION + GAMES

3 IDEAS FOR TESTING

Dice game

Yes/No game

Robocode

CI game (Jenkins)