

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green color. They are positioned diagonally, with the blue one in front of the green one.

Small QA Team Survival Strategies

Stareast 2024

Who am I and why should you listen to me?



- Over a decade of QA experience
- QA Manager at Auction Edge in Franklin, TN
- Worked at companies varying in size from small startups to international companies
- Career timeline:
 - Beta tested for Bethesda and some small indie game studios as a hobby while finishing my masters
 - Started out (officially) at an IT consulting firm as a manual tester in a team of about 30 testers
 - Moved on to a QA engineer role at a small startup where I worked solo on the development team
 - Relocated to Nashville and got an automation engineer position at an international company
 - Still have no idea how large the QA department actually was
 - Hated working at a company so large and went back to startup life, working with one other QA engineer
 - Currently managing a team of five QA engineers and hoping to expand soon



Who are you and why are you here?

- What role are you?
 - Manager or lead?
 - Developer?
 - QA?
- What's your dev to QA ratio?
 - 1 to 1?
 - 1 to 2-5?
 - 1 to 6-10?
 - 1 to 10+?
 - 0!
 - I have no idea. I've never calculated that before.

Agenda: Part 1

How to Survive



- Get organized
 - Test cases or...?
- Time management
 - Am I working on the most important thing?
- Ask for help
 - Lean on your amigos
- Communicate risk
 - CYA



Getting organized

- Get a test tool
 - Even a cheap, no frills one will do
 - Spreadsheet would probably become unmanageable over time
- Step one:
 - Create a feature map and/or test lists
 - Fast way to determine the amount of testing and scenarios needed
- Step two:
 - Record each time you execute a test scenario from your list or test part of the feature map
 - Allows you to communicate risk and coverage accurately
- Step three:
 - When you have free time (if you have any), you can add test steps to your test lists
 - This may be unneeded depending on who will be running the tests in the future



What's a feature map?

- A feature map is a visual representation of all of the actions a user can take in an application and the outcome.
- Should be done at the start of a sprint or project or else it might be too time consuming to catch up.
- Good resources on how to create one:
 - <https://johnfergusonsmart.com/feature-mapping-a-lightweight-requirements-discovery-practice-for-agile-teams/>
 - <https://johnfergusonsmart.com/feature-mapping-a-simpler-path-from-stories-to-executable-acceptance-criteria/>
 - <https://thegaconnection.com/2021/08/17/an-end-to-end-guide-about-feature-mapping/>



Time Management - Basics

- Are you working on the most important thing?
 - Ask your manager/lead
 - Ask your product owner
 - Ask for more than one top priority item
 - I like to get at least my top 4-5 priorities in case I get blocked
- What if everything is high priority?
 - Get deadlines and release dates
 - If there's no deadline or desired date, question around that to get more context. Is it really high priority?

Time Management Quadrants

	Urgent	Not Urgent
Important	I Fire Fighting Crises Pressing problems Deadline-driven projects	II Quality Time Prevention, capability improvement Relationship building Recognizing new opportunities Planning, recreation
Not Important	III Distraction Interruptions, some callers Some mail, some reports Some meetings Proximate, pressing matters Popular activities	IV Time Wasting Trivia, busy work Some mail Some phone calls Time wasters Pleasant activities



Time Management - Seeing the future

- Knowing what is coming can help you know where to allocate time
 - For example, knowing that a new version of iOS is coming out in the next few months lets you know you'll need to regression test your team's iOS app in the upcoming future, so you should get other high priority items off your plate beforehand.
- Participate in all sprint ceremonies if you can
- Build a relationship with the product team
 - Understand their vision and what their goals are



Asking for help - Manual testing

- Whole Team Quality AKA Quality is everyone's responsibility
 - Requires buy in from management
 - Contingent on these teams also not being strapped for resources
- Have to be organized before you can successfully ask for help and trust the help you get
- Ask your product team and/or customer support specialists to help test new features or obscure bugs
- Ask your developers to pair with you on more technical stories
 - Data engineers can create sql queries to assist with data validation
 - Full stack developers should be able to show you what API queries you can use to test the API itself versus manually testing API changes in the UI
 - These should be saved and used to create automation later
- If you pair with others, make sure you're engaged and asking questions!



Asking for help - Automated testing

- It might be easier and faster for the team's developers to pick up automation tasks if more technical skills do not exist in the QA department.
- Even just getting an increase in unit test coverage will increase a product's quality.
- If developers are already using tools like Postman, leverage what they already have to create integration tests.
- Make a case with management to allocate a certain amount of time to automation.

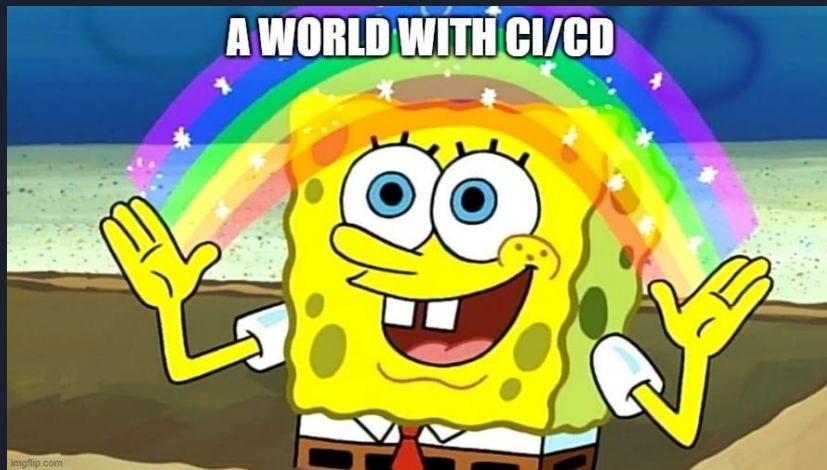
Communicating Risk



- “The product owner told me that x,y, and z are the most important features in this release and I only had time to test these features. Any other lower priority fixes/features have not been tested by the QA team. Is this a risk we can accept or should these minor changes wait?”
- “I’ve feature tested x of y number of stories in our sprint. If these all go to production, I can only certify the quality of these I tested. Does management accept that risk?”
- If management is uncomfortable (and they should be!), talk about what it would take to provide more test coverage and what level of coverage/risk is acceptable.

But what about CI/CD?

- I tested this one code change, it passed QA. It, and only it, can go to production.
- Smaller releases *usually* mean lower risk
- Requires expert knowledge of the application and how code might cause unintended regressions.





Part 2 Agenda:

How can I make the case to expand my team?

- Identify how many people you need and what skill sets
- Metrics
 - How long do stories stay in QA?
 - Is the team missing deadlines?
- Examples of bugs that were missed
- Examples of how QA could be better



How many people do you need and what skills?

- Do you need manual QA engineers or automation engineers?
- How many new hires would it take to get your dev to qa ratio to somewhere around 1 to 5?
- Is anyone on your team able to do automation?
 - Is yes, why aren't they doing it?
 - What would it take to free up their time so they can work on more technical tasks?
 - If no, what type of automation would you like your new hire(s) to do?

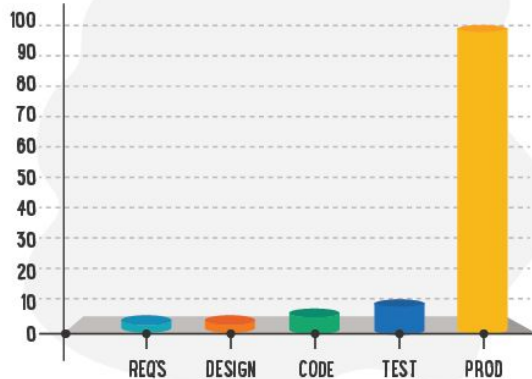


QA Metrics

- Time spent in QA status
- Number of test scenarios NOT run
- Number of defects identified
 - Break this out by priority (Critical, High, Medium, Low)
 - If this is broken out by environment, make sure to specify number found in production
- Talk to team scrum masters
 - Has velocity stagnated due to QA?
 - Jira creates reports that can support this.
- Talk to customer support
 - Get numbers on how many new tickets are being opened due to bugs
 - Ask them for any other metrics they might be able to provide to make your case

Examples of bugs that are missed

THE RELATIVE COST OF FIXING DEFECTS



ILLUSTRATED BY SEGUE TECHNOLOGIES

- Everyone has a production bug nightmare story. Most have more than one.
- Identify how many 911s your team is facing on a regular basis.
- Finding bugs via QA is cheaper than customers finding them.



Examples of how the QA department could be better

- If your dev to QA ratio is already reasonable, but you have no time for automation, writing test cases, participating in scrum ceremonies, or the many other tasks QA can do... research those and make the case for why your company needs them.
- For example, if your company has no performance tests, but customers are constantly complaining about how slow your applications are... You have a good case to either hire a manual QA engineer to free up someone's time or hire an automation engineer to do the work.



But my company doesn't want to hire...

- Is this a systemic issue?
 - Are other teams also strapped for resources?
- Is this just QA?



But my company doesn't want to hire...

- Other options:
 - Working with a local college to create an internship program.
 - Look for a third party company that could provide some support.
 - Crowd based manual testing
 - Hire a contractor
- Keep bringing it up
 - Sometimes getting a new hire is more about timing
 - Talk to your manager, HR, or recruiter to find out when the best time to pitch new hires is
 - Get others to make your case too
 - Scrum masters
 - Other managers (Dev, Product)

Questions?

Contact me at amandae@auctionedge.com or via LinkedIn

