



it's
a

Bug!

by Robert
and Ann

Testing Lessons from Labor Triage

Bug triage is about making decisions about the fate of software bugs: Should we keep a bug? Should we fix it now? Can we fix it later? Can we live with it? What other choices do we have? What should we do next?

I have implemented bug triage in every software project I've run since 1992. My inspiration in organizing bug triage comes directly from the world of labor and delivery nursing, which I learned about from my wife, Anne, a nurse at the birthing center of the Royal Victoria Hospital in Montreal. Anne has triaged hundreds of expectant mothers arriving at the hospital in anticipation of the childbirth experience.

Labor and delivery triage is about deciding the course of treatment of patients arriving at the hospital. Anne must decide the course of action on the spot, often with minimal information guiding life-critical decision making.

I apply the four basic steps of labor and delivery triage directly to bug triage:

- Preliminary Assessment
- Interview
- Exploration and Observation
- Taking Action



Preliminary Assessment— Looking the Bug Right in the Eye!

The labor triage assessment begins as soon as the triage nurse sees the patient. The triage nurse observes the patient: Can she talk? Is she agitated? Is she about to faint? Is she excited, anxious, or in any sort of trauma? Her first judgment is related to the urgency of the situation. Sometimes the patient needs immediate medical care, even before Anne knows the patient's name—the baby will not wait. The triage nurse must be prepared for anything and remain calm despite the adrenaline rush. In some cases the triage nurse delivers the baby on the spot. It's a job that has all the excitement of a TV drama. The triage nurse knows what to look for and is trained to recognize and act in response to critical situations.

The bug triage assessment begins as soon as I see the bug. I take a good look at the bug: Will critical user tasks be blocked? Is the bug infectious? Will it break other things? Does it violate laws, regulations, or contractual obligations?

Did we lose client data? Does this bug block us from doing other work? My first judgment is related to the urgency of the situation.

In the preliminary assessment of bugs I look for those bugs that demand immediate action and then I initiate whatever course of action is required. In some projects I call this bug filtering. I cut out any paperwork or bureaucracy involved and get right to resolving the bug. When a bug demands action, I take action.

To hone my preliminary assessment skills I have to study a lot of bugs. I want to avoid crying wolf and getting developers to start fixing a problem that is not really urgent. Knowledge gleaned from studying bug taxonomies is a great source of information. I always study real bugs experienced in similar projects. In order to get a good sense of the urgency of a problem, I try to understand the impact on the end-user of not fixing the problem immediately.

The preliminary assessment helps trigger immediate action before going to bug review meetings or doing any further testing. Being prepared for a preliminary assessment includes a blend of knowing what to look for and knowing how to get things done. I make sure project managers, development leads, and all team stakeholders know that I do preliminary assessments and that I can sidestep bureaucracy in some urgent cases.



Interview— Understanding Context

In labor triage context is everything. The same conditions observed with a different context can lead to dramatically different results. One example is that of labor contraction timing. Imagine that a patient calls triage reporting contractions lasting sixty to ninety seconds with mild intensity and taking place every ninety minutes. If you knew that the mother had recently experienced a car accident, you would want to see her right away. If you knew the mother was just watching a soap opera on TV, you might suggest she call back in a few hours and take it easy.

The triage nurse interviews the patient to learn about context. She asks questions that help identify the phase of labor the woman is in. Some basic information about the patient is collected: name,

TRIAGE NURSING NOTES

A principle used by triage nurses is “If you do not see it in writing, it was not done.” Note taking is a critical skill of the triage nurse. Almost any medical professional who will be involved in the case may reference these nursing notes. A triage note includes entries for each and every event, interaction, observation, test done, and test result. Notes include the questions asked, responses given, and action taken by the nurse. Each notation includes the date and time the event took place. The triage nurse must take detailed notes on the entire clinical encounter coupled with all test results and observations. Experienced practitioners use a terse, unambiguous style to capture their notes. They use a template form to guide note taking, which focuses on the observations and actions taken but does not include subjective assessment—as Sergeant Friday would say, “Just the facts.”

NOTE TAKING IN TESTING

I have learned over the years that communication is one of the most important skills in testing. Imagine the audience for a bug report: We are writing to other testers, test leads, development leads, help desk staff, developers, product managers, technical writers, project managers, and many different product stakeholders. It is important for testers to meaningfully communicate to this varied audience without over simplifying, confusing, or complicating the information. Software testing session notes and bug reports should be able to stand up to the same level of scrutiny as triage nursing notes.

age, gestational age, doctor, any previous pregnancies, and what happened during those pregnancies. In addition, some information about the pregnancy is collected, such as any special conditions, results of ultrasounds, and information about any medical interventions. Information about the actual condition of the patient is collected: frequency of contractions, whether the “water” has broken, and any special pains or indicators. The interview takes only a few minutes and provides information that is used to decide if the patient should be admitted, observed for a few hours, or sent to another department.

In bug triage context is everything. The same bug may require urgent intervention in one context and be easily deferred or worked around in another context.

Testers interview bugs all the time. Testers build up information to help decide which bugs to fix and which bugs to keep. They collect basic information in a bug-tracking system: When did it occur? What version of the software was being used? What operating system? What build, locale, state of the database, and state of the system? What else was going on at the same time? Testers ask questions about the specifics that exposed the problem: Does it happen all the time? What steps could reproduce it? What other tests related to the problem have been done and what were the results? Other questions relate to the condition of the bug: What is the severity? What is the consequence of not fixing the problem? How much damage has the bug caused?

I consider the following three sources of context information about the bug before taking action:

Business context: Why is the bug of importance to our business? What would the impact be if the bug were not fixed? Would workarounds be acceptable?

Technical context: Are there any special technical concerns about the bug? Is it in our code? Do we depend on a third party? Could fixing this bug break something else?

Organization context: Was the issue reported as part of testing or from the field? Will there be further levels of testing downstream? Can we gracefully

update the client after deployment? Do we have access to developers who can fix it?



Exploration and Observation—Learn More About It

After the interview the triage nurse performs some medical tests to learn more about the patient’s condition. The triage nurse checks body temperature, blood pressure, and fetal heart rate and does some basic blood and fluid tests. These test results combine with interview and preliminary assessment data to help guide decision making. When conditions are uncertain, the triage nurse will monitor the patient for a few hours. Monitoring uses different medical testing techniques, such as ultrasounds, to help observe emergent behaviors before a medical course of action is taken.

Sometimes I need more information before deciding what to do with a bug.

I may assign a tester to further investigate the problem or to work directly with developers to get a better understanding of the bug. Exploratory testing around the problem area can be used to gather additional data to help guide decisions. Are there other ways to trigger the bug? Are there other emergent behaviors associated with the problem?

I encourage testers to capture data about the software being tested and to observe the environment in which the software is running. It may be important to measure how much CPU capacity is being used by the application under test. How fast is the application responding to requests? Do we have basic data integrity? How are systems resources consumed?

I also want to confirm that other parts of the application work well enough to handle typical transactions. When a bug shows up in one area of the software, it is important to confirm quickly that other parts of the code or data are still working.



Taking Action—Getting Things Done

Three outcomes may result from labor triage: The mother is admitted, observed, or sent home.

The triage nurse has access to established medical protocols to help her decide appropriate actions based on the data collected during the preliminary assessment, interview, and clinical observations. The protocols are described in a clear one-page format in which the presenting condition, key context drivers, and recommended course of action are all spelled out. The triage nurse does not rely on the protocol manual when faced with the day-to-day realities of labor triage. She is in the hot seat and must react to the realities of the situations with which she is confronted. She must act. She must combine her experience and knowledge on the fly.

I triage bugs to determine one of three possible outcomes: Fix it now, fix it later, or do not fix it.

I want to make sure all bug triage decisions are influenced by business, technical, and testing factors. I have never been able to put together a crisp series of protocols such as those used in labor triage. I make sure that a small team of stakeholders makes decisions about bug priority. I like to involve a product manager who advocates for the customer, a development lead who is aware of the

technical risks of the project, and a test lead who is driving the testing initiative of the project. Generally the test lead provides objective information about the bug from the preliminary assessment, interview, and exploration steps. The team weighs business and technical concerns in order to come up with a decision about the bug.

This final decision draws upon all of the information gathered so far. I find bug triage teams work best when the team is 100 percent in sync regarding the purpose of the project. Why are we doing this project? What are the key business issues? What are the technical challenges? What value does this project offer and to whom? If team members are in “value sync,” then they will be better able to make difficult decisions.

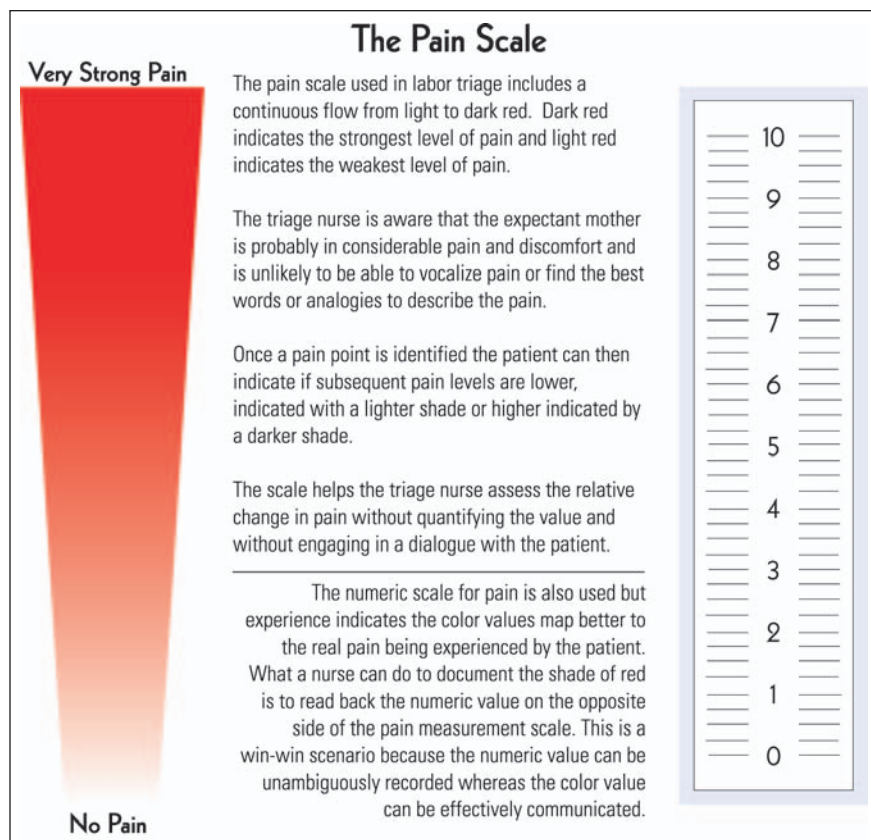
I ask the team to consider these important questions for each bug: What is the benefit of fixing the bug? What is the consequence of not fixing the bug? I also want to make sure they consider the reverse: What is the *consequence* of fixing the bug? What is the *benefit* of not fixing the bug? The team considers the tradeoffs related to fixing the bug or leaving it in there.

QUANTIFYING PAIN

The triage nurse has many interesting methods to help quantify factors of the pregnancy that would otherwise be very difficult to describe. One tool is a color-coded pain scale that includes a range of shades of red presented from light red to very dark red. The color system is used when patients are asked to describe their level of pain. Although this does not give an absolute measure, it is very effective in helping patients communicate when pain levels are increasing or decreasing as time progresses or as a result of different medical interventions.

QUANTIFYING SEVERITY

One of my most difficult problems is trying to find a meaningful way to quantify the severity of a bug. How much damage could the bug cause? I have used many schemes over the years, and on a recent project I took the labor triage pain scale to a customer site. Developers, testers, and product managers would point to the shade of red associated with a bug to indicate the severity. This let us immediately see disagreements between project stakeholders, which subsequently led to better understanding. We also could see the relative severity of different bugs. Color coding severity may appear subjective but it definitely helped us focus and communicate more effectively!





Final Words

In *Dynamics of Software Development*, Jim McCarthy suggests that project teams should “triage ruthlessly” to make all decisions that shape a product. Although they are dramatically different domains, labor triage offers a lot of valuable lessons that we can apply to software testing projects. Testers can model some of their workflow based on the stages and activities the triage nurse uses to guide decisions.

Just as a labor triage nurse must know when to send a patient home, when to start treatment, and when more knowledge is needed, the tester should learn to decide which bugs to fix, which bugs to keep, and when to get more information before making a decision. Triage helps testers react and adapt to the critical context drivers on our testing projects and helps us focus on delivering the value that makes a difference. **{end}**

WHEN IT'S TIME



TO OUTSOURCE SOFTWARE TESTING



IT'S TIME TO CALL **ACULIS**.

ON-SITE, OFF-SITE, OFF-SHORE
A BLENDED APPROACH TO SOFTWARE TESTING.

WITH STATE-OF-THE-ART FACILITIES IN THE U.S. & ARGENTINA

HOW DOES YOUR QA PROCESS RATE?

Visit Us at StarEast 2008 - May 7th & 8th - **Booth #45**

For a Chance to Win a **FREE** 3-Day QA Assessment & Scorecard Report



ACULIS / **IT ACCELERATOR**
DEVELOP IT / TEST IT / GLOBALIZE IT / STAFF IT

1-866-4ACULIS / WWW.ACULIS.COM

PATIENT ADVOCACY IN TRIAGE NURSING

All expectant mothers do not need the same blend of tests, interventions, and medical care. The triage nurse does not just kick off processes. The triage nurse also actively represents the patient to the other practitioners called upon to consult or act in the case. The triage nurse gives other medical professionals a heads up on the upcoming delivery. The triage nurse communicates the relative urgency of the patient's situation and ensures that everyone knows what may be required. In many ways the triage nurse must be aware of the work required by all medical professionals involved in the delivery process. Miscommunication can result in serious and potentially fatal problems. Triage nurses cannot exaggerate situations to draw too much attention to a case. To succeed the triage nurse must be credible. Consistency in objectively identifying the severity and priority of the case leads to this credibility. This earned credibility then triggers people to prepare and act based on the request of the triage nurse. Without credibility the triage nurse would be ineffective because nobody would listen.

BUG ADVOCACY IN TESTING

Testers without credibility find it difficult to influence projects. Testers build credibility in many ways. Testers should not exaggerate the potential impact of bugs, nor should they trivialize bugs. Testers can advocate bugs effectively by consistently providing clear and objective input about the issue being described. Testers should anticipate the type of information needed by all members of the development team. Just showing that the bug exists may be insufficient. Testers can provide information that may be useful in evaluating the priority of the bug, such as the consequence to the user community if the problem is not corrected. Testers can provide information that may be useful to the developer such as the technical state of the application and actions that took place before the bug was observed. The tester can anticipate the type of questions that may arrive from the help desk or customer support departments. The tester can provide a pithy summary of the bug, which helps non-technical executive stakeholders understand the nature of the concern.