Course "Debugging"

Debugging Rules 8 & 9

Prof. Dr. Lutz Prechelt
Freie Universität Berlin, Institut für Informatik
http://www.inf.fu-berlin.de/inst/ag-se/

- Rule 8:
  - Get a fresh view
- Rule 9:
  - If you didn't fix it, it ain't fixed
- Summary
- Exercise

The nine rules

1. Understand the system
2. Make it fail
3. Quit thinking and look
4. Divide and conquer
5. Change one thing at a time
6. Keep an audit trail
7. Check the plug
8. Get a fresh view
9. If you don't fix it, it ain't fixed

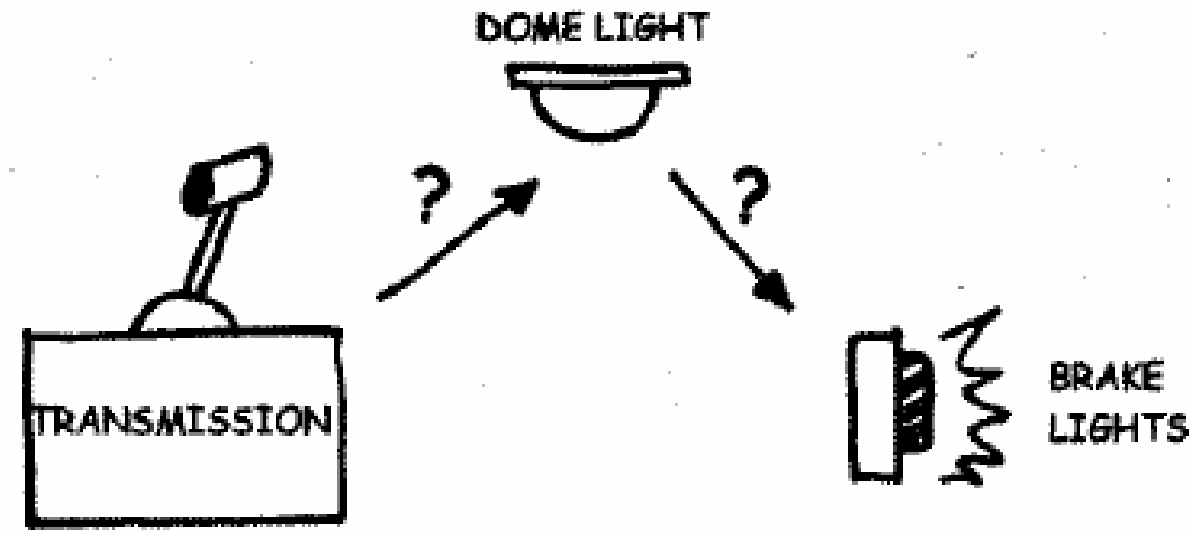
Rule 8: Get a fresh view

- "Nothing clears up a case so much as stating it to another person."

  Sherlock Holmes

Get a fresh view: Brake light war story

- A car blew the fuse of the brake lights whenever you put the transmission into reverse gear
- Several attempts made it clear that this was repeatable
- Reason???
- Owner mentioned the problem to somebody familiar with repairing this brand of car
- Immediate answer: "The dome light is pinching a wire against the frame of the car. Insulate that wire and you will be fine."



Get a fresh view: What would have happened?

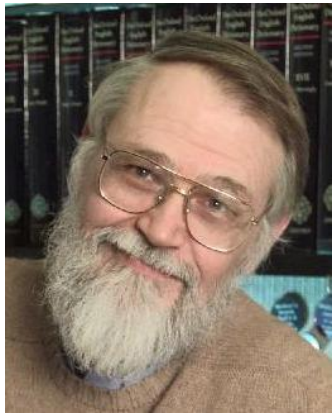
Freie Universität Berlin

What would have happened if one had applied only the other rules?:

- Understand the system
  - Obtain(!) and study car wiring diagrams
- Make it fail ; Quit thinking and look
  - He did this alright
- Divide and conquer
  - Rip out the car's wiring in parts?
  - Or obtain measurements throughout the wiring?
- Check the plug
  - There was no assumption that could be questioned

Get a fresh view: Ask an expert

Method 1 of getting a fresh view:
Ask an expert on the subject matter



- This avoids having to "Understand the system"
  - Which could be a gigantic effort in complex cases
- Make sure you can tell a real expert from a charlatan
  - Incomprehensible answers often indicate charlatans
  - Vague answers may as well
- In many cases, experts may be external consultants
  - They are expensive
  - They may or may not be worth the price

Get a fresh view: The voice of experience

Freie Universität Berlin

Method 2 of getting a fresh view:
Ask someone who has had such a situation before

- preferably many times

- Note that an experienced person need not be an expert
  - The experienced person need only know the cause (or even only the solution)
  - S/he needs not "understand the system"
  - See the war story above:
    The "expert" may have known nothing about the specific wiring of this car, except this one problem

- Note that software-experienced persons often <u>did</u> have the problem before
  - but have forgotten the cause and solution

Get a fresh view: A breath of fresh insight

Freie Universität Berlin

Method 3 of getting a fresh view:
Explain your problem to someone completely unrelated

- Advantage 1: This person is not as biased (or at least differently biased) as to where the problem should be
  - If s/he does not have useful advice or questions to contribute, s/he may at least console you
- Advantage 2: Even just explaining the problem may give you a fresh view and an insight
  - It often even works to explain the problem to your pen, the leg of your table, or your favorite rag doll

Get a fresh view: Sources

Freie Universität Berlin

Where to find expertise, experience, or insight:

- Colleagues:
  - can provide all three
- Vendor support (telephone, online):
  - usually expertise
  - often experience (perhaps via other users: online forum)
- Knowledge management systems, the Web:
  - perhaps experience,
  - often a pointer to a person with expertise/experience
- Documentation:
  - perhaps insight, always expertise (though both unspecific)
  - perhaps experience (troubleshooting guide)
- Anybody/yourself: insight by explaining it all

Get a fresh view: Subrules

- Subrule: Don't be proud
  - Asking someone is not a sign of weakness (if you've done your part before), but rather of good judgement
- Subrule: Do not assume the expert is a god
  - Mistrust expert judgement just like your own
- Subrule: Report symptoms, not theories
  - or you will reduce your chances of getting new insight
  - And if you are the helper: Don't get poisoned. Cover your ears and loudly sing LA-LA-LA-LA-LA.
- Subrule: Include observations you have not understood
  - If it is confusing for you, it may be just where somebody else can help

The nine rules

1. Understand the system
2. Make it fail
3. Quit thinking and look
4. Divide and conquer
5. Change one thing at a time
6. Keep an audit trail
7. Check the plug
8. Get a fresh view
9. If you don't fix it, it ain't fixed

Rule 9: If you don't fix it, it ain't fixed

Freie Universität Berlin

- "It is stupidity rather than courage
  to refuse to recognize danger
  when it is close upon you."

  Sherlock Holmes

If you don't fix it, ...: Used car war story

Freie Universität Berlin

After buying a used car:

- Event 1: While going up a hill, suddenly the engine stopped
  - After stopping in the breakdown lane, the motor started again at the first attempt
  - During the slow drive up the rest of the hill, the car did not fail again
- Event 2: After filling up at a dubious little gas station on a bitter-cold day, the engine stopped again while going up a hill
  - Again, it started again at the first attempt
  - Driver thought: Maybe water in the fuel line.
    Applied drygas spray.

If you don't fix it, ...: Used car war story (2)

- Event 3: Engine stopped while driving fast on a perfectly flat road
  - Did not start on first attempt; but did start on the second
- Experimentation found that the engine would stop after going at more than 80 km/h for some short while
- Driver took the car to a repair shop
  - They replaced some wires and told him it was an electrical problem. Cost: 75 Dollars.
  - The car failed again the next day; just as before
- Idea: Maybe the carburettor does not receive enough fuel in high-load situations?
  - Get a fresh view: Asked a colleague at work. Answer: "Dirty fuel filter."
  - The repair cost 50 cents.

If you don't fix it, ...: Subrules

- Subrule: Check that it's really fixed
  - You made it fail before, did you? Try again.
- Subrule: Check that it's really your fix that fixed it
  - After checking that it was fixed, take out your fix and make it fail
  - Note: Sometimes this is unecessary, too risky, or too cumbersome
- Subrule: It never just goes away by itself
  - If it stops failing without a proper fix, put in instrumentation to understand the failure next time
  - or analyze the differences to the failing version if you can
- Subrule: Fix the cause
  - Look behind the first-level cause of the failure and try fixing its root cause.

If you don't fix it, ...: Transformer war story

Freie Universität Berlin

- Author was given a used integrated stero set (tuner, amplifier, 8-track tape deck)
  - Did not work at all
  - Measurements found: No output from the power supply
- Author ordered a new transformer from the vendor
  - Waited months before it arrived
- Replaced the old one and turned on the radio: Music!
  - until an hour later, the transformer goes up in smoke
- Solution: The tape deck had a short circuit
  - The fix (new transformer) had repaired only the symptom (broken transformer) not the cause.
  - Violated: Understand the system; quit thinking and look. Misapplied: Make it fail.

If you don't fix it, ...: Fix the process

- Fixing the root cause often does not just touch the product
- Rather, it goes right into the production process
- Example:
  - Problem: Oil on the factory floor
  - Bad fix: Wipe the oil up
  - Better fix: Also repair the leaky fitting of the machine that dripped the oil
  - Still better fix: Vibrations wrecked the machine's fitting. Also use two more bolts to mount the machine to the floor.
    - Are there other similar machines?
  - Still better fix: Also change the vibration evaluation process used during the design of machine bolting configurations

The nine rules: Summing up again

1. Understand the system
2. Make it fail
3. Quit thinking and look
4. Divide and conquer
5. Change one thing at a time
6. Keep an audit trail
7. Check the plug
8. Get a fresh view
9. If you don't fix it, it ain't fixed

1. Understand the system

- Read the manual
- Read everything in depth
- Know the fundamentals
- Know the mechanisms
- Understand your tools
- Look up the details

2. Make it fail

- Do it again
- Start at the beginning
- Stimulate the failure
- But do not simulate the failure
- Find the uncontrolled condition that makes it intermittent
- Record everything and find the failure signature
- Do not trust statistics too much
- Know that "that" can happen
- Never throw away a debugging tool

3. Quit thinking and look

- See the failure
- See the details
- Build instrumentation in
- Add instrumentation on
- Don't be afraid to dive in
- Watch out for Heisenberg
- Guess only to focus the search

4. Divide and conquer

- Narrow the search with successive approximation
- Find the right range
- Determine which side of the failure you are on
- Use easy-to-spot test patterns
- Start with the bad parts
- Fix all defects you learn (or know) about
- Fix the noise first (just don't get carried away)

5. Change one thing at a time

- Isolate the key factor
- Grab the brass bar with both hands
- Change one test at a time
- Compare it with a good one
- What have you changed since the last time it worked?

6. Keep an audit trail

- Write down what you did, in what order, and what happened as a result
- Understand that any detail could be the important one
- Correlate events
- Understand that audit trails of design are also good for debugging
- W R I T E I T D O W N !

7. Check the plug

- Question your assumptions
- Start at the beginning
- Test the tool

8. Get a fresh view

- Ask for fresh insights
- Tap expertise
- Listen to the voice of experience
- Know that help is all around you
- Don't be proud
- Report symptoms, not theories
- Realize that you don't have to be sure

9. If you don't fix it, it ain't fixed

- Check that it's really fixed
- Check that it's really your fix that fixed it
- Be aware that it never just goes away by itself
- Fix the cause
- Fix the process

Small virtual debugging exercise

- Assume you move into a rather old house
- While cleaning on the first day, you tried to use the vacuum cleaner in the living room, but
  - when you switched it on, it did not start.
  - Instead, the room lights turned on.
- What do you do?

- **Now I am the house.**
- **Debug me!**

Remark: When at a help desk

For user-driven remote debugging situations:

- Follow the rules
  - despite the unenlightened user
- Verify actions and results
  - Misunderstandings of your instructions are very likely
- Use automated tools
  - Take the user out of the loop whenever you can
- Verify even the simplest assumptions
  - You cannot rely on anything
- Use available troubleshooting guides
- Contribute to troubleshooting guides

What is difficult when trying to apply the rules

Freie Universität Berlin

...and what to do about it

- 1. Understand the system
  - Requires a lot of time
    - Learn it piece-by-piece; don't tell your boss
  - Requires a lot of interest and patience
    - Most systems contain a lot of great ideas; understand that understanding them is fun
  - May be just impossible
    - The thrill of software engineering: Manage anyway

What is difficult... (2)

- 2. Make it fail
  - May be arbitrarily difficult and the effort is unknown in advance
    - Invest not more than finding the defect would be worth
    - Use enough of your effort for building real good instrumentation (so you catch it next time it occurs)
- 3. Quit thinking and look
  - Pride gets in the way
    - Work on your personality
- 4. Divide and conquer
  - Some hypotheses are <u>much</u> harder to test than others
    - Invent alternative checking methods (perhaps partial ones)
    - Find cheaper surrogate hypotheses that are almost as useful

What is difficult... (3)

- 5. Change one thing at a time
  - Requires patience
    - Automate
    - Keep an audit trail and analyze how much it hurts to violate the rule
- 6. Keep an audit trail
  - Requires discipline and patience
    - Automate
    - Get help (pair debugging)
- 7. Check the plug
  - You may be unaware of your basic assumptions
    - Use brainstorming (with colleagues) to form a checklist

What is difficult... (4)

- 8. Get a fresh view
  - You may be too deep into your problem to understand the fresh view you got
    - Take some time off
- 9. If you don't fix it, it ain't fixed
  - Task pressure demands not working on it if it does not appear broken
    - This may even be acceptable
    - But keep some notes of what you learned so far
- ALWAYS:
  - You may be too ignorant to see that (or where or how) you can apply a rule
    - Talk to a colleague in intervals when debugging takes long

Summary: The nine rules

1. Understand the system
   - even if that is hard
2. Make it fail
   - even if that is hard
3. Quit thinking and look
   - even if you think you know what is the matter
4. Divide and conquer
   - and do not jump to conclusions
5. Change one thing at a time
   - even if it seems too simple
6. Keep an audit trail
   - in writing!
7. Check the plug
   - at least after a while
8. Get a fresh view
   - at least after a while
9. If you don't fix it, it ain't fixed
   - so fix it and make sure

Thank you!