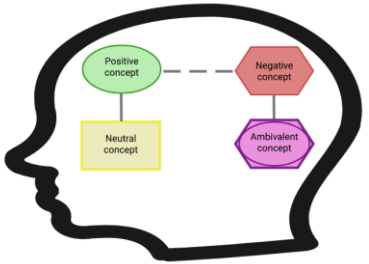
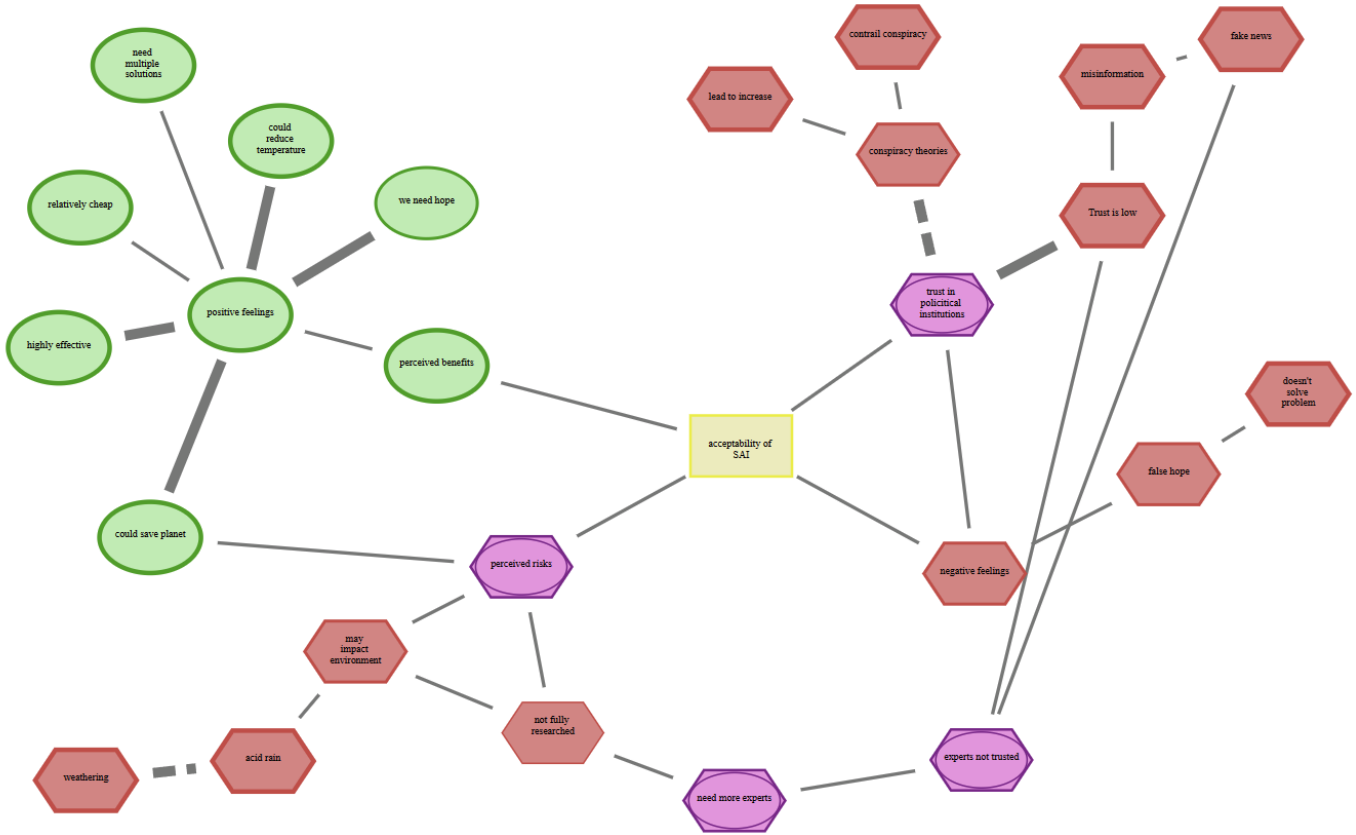
Software Testing

to improve the Cognitive-Affective Map tools



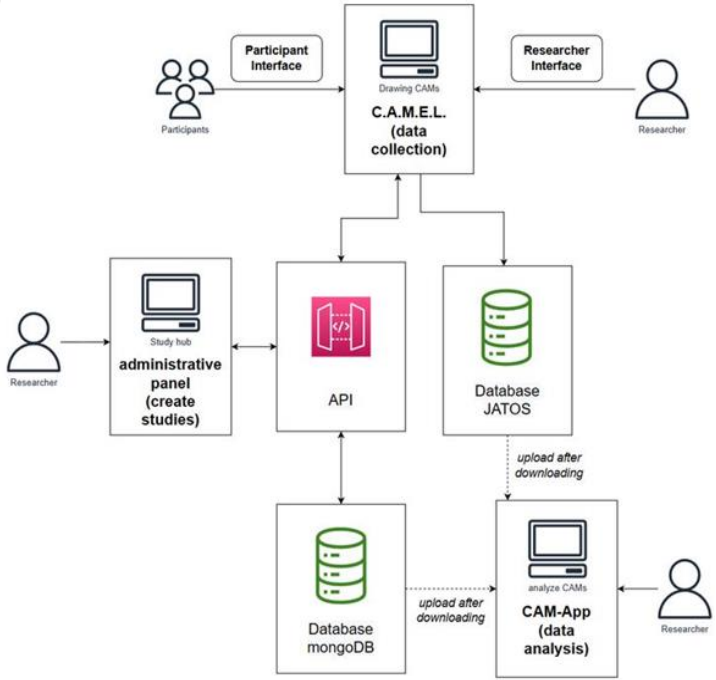
Cognitive-Affective Maps (CAMs)



Fenn, J., Helm, J. F., Höfele, P., Kulbe, L., Ernst, A., & Kiesel, A. (2023). Identifying key-psychological factors influencing the acceptance of yet emerging technologies–A multi-method-approach to inform climate policy. *PLOS Climate*, *2*(6), 1–25. https://doi.org/10.1371/journal.pclm.0000207

Developed tools for Cognitive-Affective Mapping

- Central web page: https://drawyourminds.de
  - General information on the developed tools; possible to set-up CAM studies without the need for programming
- Online documentation: https://osf.io/q5hj4/
  - Detailed documentation of developed CAM tools and explanations for how to set up studies with multiple examples; thematically sorted CAM literature
- GitHub: https://github.com/Camel-app
  - All the code of the programmed CAM tools:
    - Data Collection: Cognitive-Affective Map Extended Logic (C.A.M.E.L.)
    - Data Analysis: CAM-App
    - Web page



Our goal

- **Software debugging** of developed CAM tools
  - Data Collection: Cognitive-Affective Map Extended Logic (C.A.M.E.L.)
  - Data Analysis: CAM-App
  - administrative panel (webpage)

> after debugging and improving these tools it will be finally deployed on the bwCloud

- Collection of suggestions for possible improvements and extensions of the tools

Being a great software tester

You are a valiant knight combating bugs and glitches to protect the integrity of software.

Be a

- Hero
- Embrace your critical role

(OpenAI, 2024)

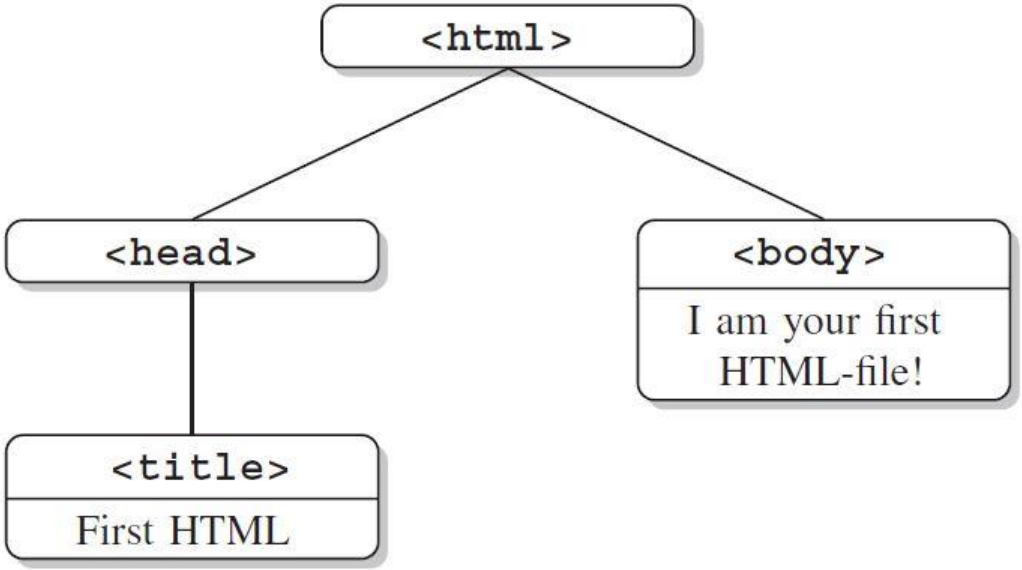
Technical fundamentals of browsers

Building blocks of web pages

Technology	Role in Web Development	Key Characteristics	Examples
HTML (HyperText Markup Language)	Defines the structure and content of web pages.	- Uses tags to denote elements such as headings, paragraphs, links; forms the skeleton of the web page.	`<h1>Title</h1>` `<p>This is a paragraph.</p>`
CSS (Cascading Style Sheets)	Styles the visual presentation of web pages.	- Controls layout, colors, fonts. - Separate from HTML to allow design changes without altering the structure.	`<p style="color:blue">This is a paragraph</p>`
JavaScript	Adds interactivity and dynamic content to web pages.	- Can manipulate both HTML and CSS. - Enables responsive and interactive web elements.	`document.getElementById("demo").innerHTML = "Hello JavaScript!";` `alert("Hello World");`

.html: tree-structure

```
<!DOCTYPE html>
<html>
<head>
  <title>First HTML</title>
</head>
<body>
  I am your first HTML file!
</body>
</html>
```



From Munzert, S., Rubba, C., Meißner, P., & Nyhuis, D. (2015). *Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining*. John Wiley & Sons; Figure 2.5.

.html

Hypertext Markup Language, first proposed by Tim Berners-Lee (1989), central agency: World Wide Web Consortium

```
<html>
<head>
    <title>First HTML</title>
</head>

<header>
    <h1> Hello World </h1>
</header>
<body>
    I am your first HTML-file!
</body>
<footer>
    Bye bye
</footer>
</html>
```

.css

Cascading Style Sheets: a language for describing the layout of HTML (and other markup documents)

```
<html>
<head>
    <title>First HTML</title>
</head>

<header>
    <h1 style="font-size: 22px;"> Hello World </h1>
</header>
<body>
    I am your first HTML-file!
</body>
<footer style="background-color: red;">
    Bye bye
</footer>
</html>
```

.js

JavaScript: allows the browser to change the content and structure of the document after it has been loaded from the server, enabling user interaction and event handling

```
<html>
<head>
    <title>First HTML</title>
</head>

<header>
    <h1 style="font-size: 22px;"> Hello World </h1>
</header>
<body>
    I am your first HTML-file!
</body>
<footer style="background-color: red;">
    Bye bye
</footer>
</html>
```


```
document.querySelector('footer').addEventListener("click", () => {
    alert('IEEEE you clicked on the footer!');
})
```

.js: DOM event types

- mouse events: mousedown, mouseup, click, dblclick, mousemove, mouseover, mousewheel, mouseout, contextmenu
- touch events: touchstart, touchmove, touchend, touchcancel
- keyboard events: keydown, keypress, keyup
- form events: focus, blur, change, submit
- window events: scroll, resize, hashchange, load, unload
- …

used eventListeners to build up the interactive parts for users:
https://camgalaxy.github.io/

Using JavaScript in Your Browser

Step 1: Open Your Browser's Developer Console

- **Instructions:**
  - For Chrome, Firefox, or Edge: Press `Ctrl+Shift+J` (Windows/Linux) or `Cmd+Option+J` (Mac).
  - For Safari: Enable the Develop menu in Safari's Advanced preferences, then press `Cmd+Option+C`.
- **Purpose:** The console is a tool that developers use to log information as part of the JavaScript development process. It's also where you can run JavaScript code directly.

Step 2: Write or Paste Your JavaScript Code

- **Instructions:** Click into the console, type your JavaScript code, or paste code you've copied. For example, try `console.log('Hello, world!');`.
- **Purpose:** To see how JavaScript can output data to the console. This is a basic way to start interacting with JavaScript.

Step 3: Execute Your Code

- **Instructions:** Press `Enter` to run your code.
- **Purpose:** This step runs your JavaScript code. If you used the example code, you should see "Hello, world!" printed in the console.

Hands on technical fundamentals of browsers

to get a basic understanding of the technical fundamentals of browsers

- Play around on http://example.org/

- all shown examples are on GitHub:
  - https://github.com/FennStatistics/Project_testingCAMtools/tree/main/hands%20on%20browser
- if you want to play around with the shown examples you could, install Visual Studio Code (https://code.visualstudio.com/) and the "live server" extension and click on "Go Live" at the bottom right

Learning Materials - Web Development

using programming to innovate, solve problems, and harness technology's potential (picture created by OpenAI, 2024)

Online Courses:

- freeCodeCamp [free]: https://www.freecodecamp.org/
- codecademy: https://www.codecademy.com/catalog
- udemy - Modern JavaScript: https://www.udemy.com/course/modern-javascript-from-novice-to-ninja/

Online Documentations / AI:

- W3Schools: https://www.w3schools.com/html/default.asp
- ChatGPT: https://openai.com/chatgpt

Learning Materials - Web Development

using programming to innovate, solve problems, and harness technology's potential (picture created by OpenAI, 2024)

YouTube Channels:

- Web Dev Simplified: https://www.youtube.com/@WebDevSimplified
- Net Ninja: https://www.youtube.com/@NetNinja
- Fireship: https://www.youtube.com/@Fireship
- Dani Krossing: https://www.youtube.com/@Dani_Krossing

Books:

- Haverbeke, M. (2024). *Eloquent JavaScript, 4rd Edition: A Modern Introduction to Programming*. No Starch Press. https://eloquentjavascript.net/

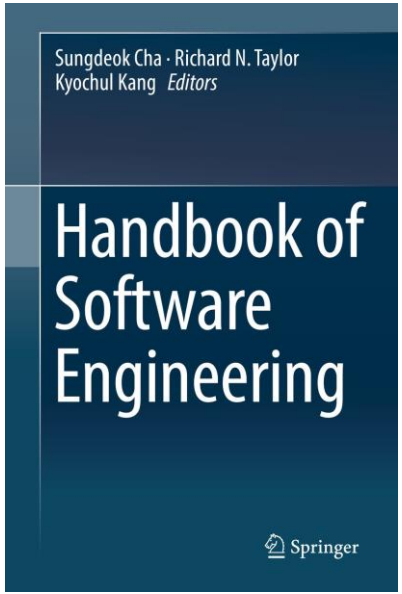
Importance?!

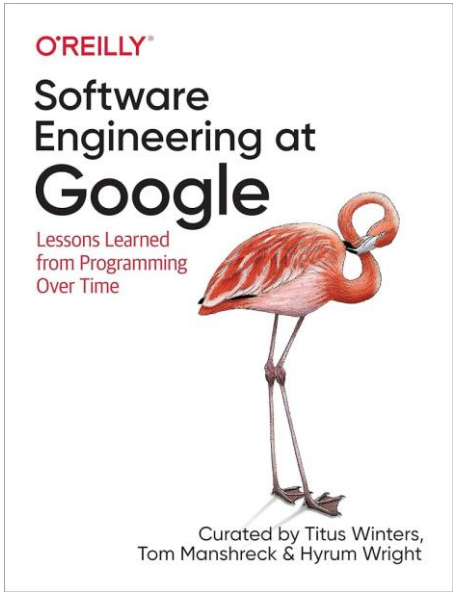
Static vs. Dynamic Web Pages

Feature	Static Web Pages	Dynamic Web Pages
Content	Fixed content	Content changes dynamically
Load Time	Generally faster	Can be slower due to processing
User Interaction	Limited interactivity	Highly interactive
Technologies	HTML, CSS	HTML, CSS, JavaScript, Server-side languages

- **price discrimination**
- **products recommendation**; Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, *7*(1), 76–80. https://doi.org/10.1109/MIC.2003.1167344

Software testing





Software Engineering vs. Programming

Aspect	Software Engineering	Programming
Definition	A systematic, disciplined, and quantifiable approach to the development, operation, and maintenance of software.	The process of writing, testing, and maintaining the code in a software project.
Scope	Broad, covering the entire software development lifecycle including planning, management, design, development, testing, and maintenance.	Focused on the implementation phase, specifically the creation of code that meets software design specifications.
Objective	To ensure software is developed in a cost-effective, timely, and high-quality manner to meet or exceed stakeholder expectations.	To translate software design and specifications into operational software by writing and testing code.
Key Activities	Requirements analysis, system design, software architecture, project management, quality assurance, and maintenance.	Writing code, debugging, and performing unit testing to ensure functionality meets specified requirements.
Outcome	Delivering a complete software system that is efficient, reliable, and meets user needs.	Producing functional software components and modules as part of the larger system.

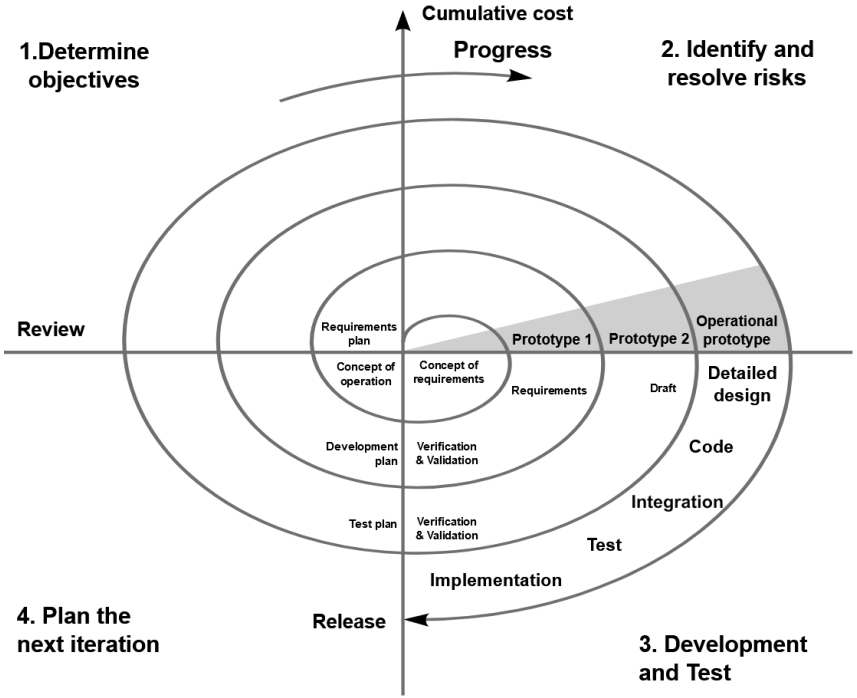
Phases of the Rational Unified Process

1. **Inception Phase**
   - Goal: Define the project's scope and business case.
   - Milestone: Lifecycle Objective Milestone (Project viability).
2. **Elaboration Phase**
   - Goal: Analyze project needs in detail and establish a solid architectural foundation.
   - Milestone: Lifecycle Architecture Milestone (Architectural baseline).
3. **Construction Phase**
   - Goal: Develop and test the software product iteratively.
   - Milestone: Initial Operational Capability Milestone (Product readiness).
4. **Transition Phase**
   - Goal: Deploy the software in the user environment and ensure stakeholder satisfaction.
   - Milestone: Product Release Milestone (Transition to maintenance).

(Ambler, 2005)

The Incremental Commitment Spiral Model

- risk-driven process model
  - **cyclic approach**: repeating process that gradually adds more detail and functionality to a system, making it clearer and more complete over time while reducing potential problems or risks
  - **milestones**: key achievement markers to make sure everyone involved agrees on realistic and satisfactory ways to build the system



From Boehm, B. (1988). *Spiral model*. File:Spiralmodel_nach_Boehm.png, Spiral_model_(Boehm,_1988).png. https://commons.wikimedia.org/wiki/File:Spiral_model_(Boehm,_1988).svg

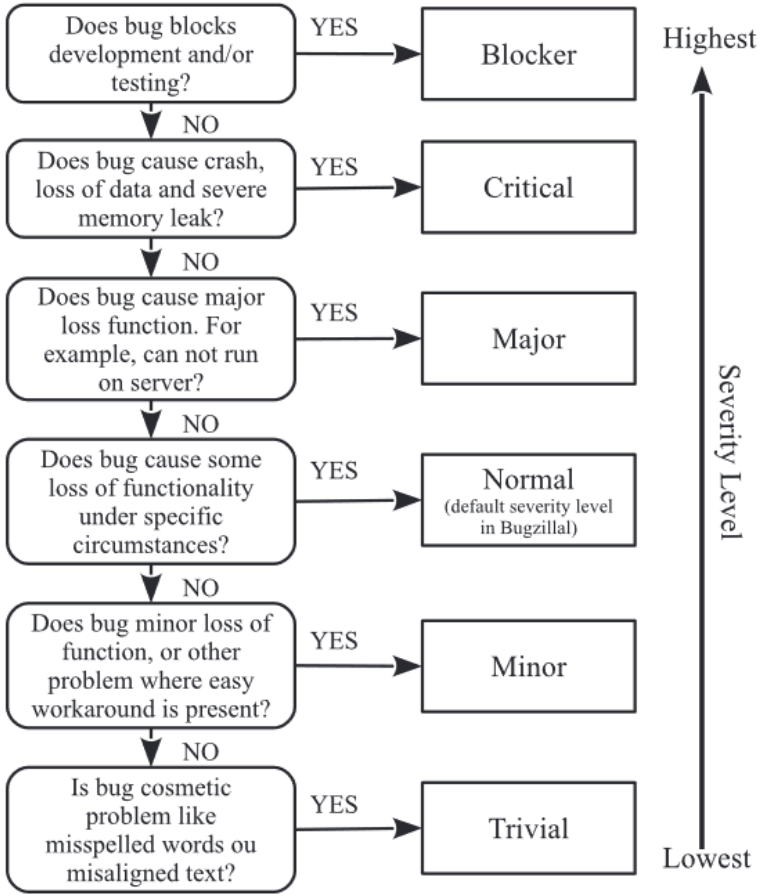
(Boehm & Hansen, 2001)

Testing - with a focus on bug reports

Bugs

In the context of software testing, a "bug" refers to an error, flaw, or fault in a software program that causes it to produce an incorrect or unexpected result, or to behave in unintended ways.

(OpenAI, 2024)



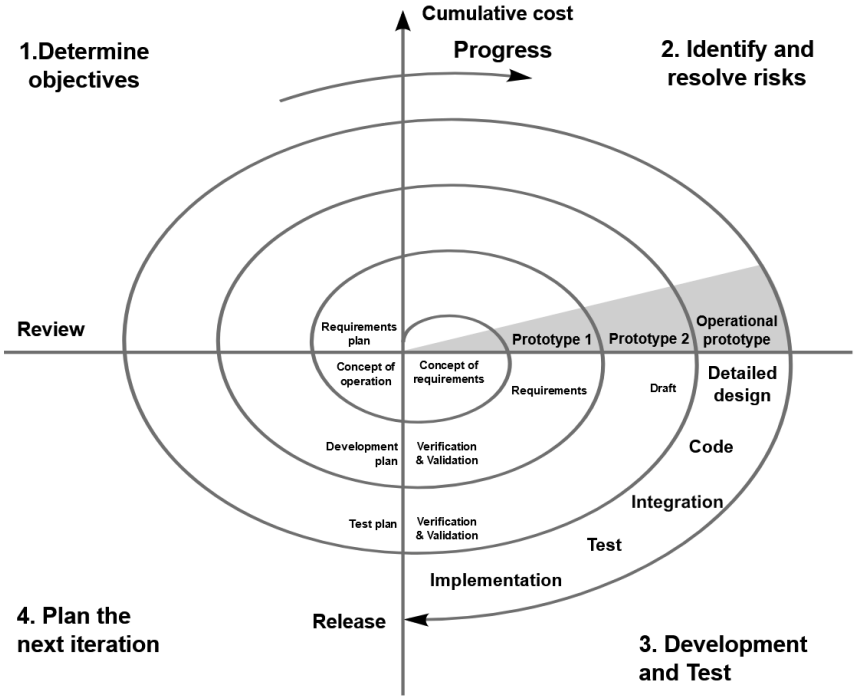
From Gomes, L. A. F., Torres, R. da S., & Côrtes, M. L. (2019). Bug report severity level prediction in open source software: A survey and research opportunities. *Information and Software Technology*, *115*, 58–78. https://doi.org/10.1016/j.infsof.2019.07.009; Figure 2 (b)

The Incremental Commitment Spiral Model is real

- Maintaining software is an ongoing activity.

e.g., see lab.js releases:

https://github.com/FelixHenninger/lab.js/releases



From Boehm, B. (1988). *Spiral model*. File:Spiralmodel_nach_Boehm.png, Spiral_model_(Boehm,_1988).png. https://commons.wikimedia.org/wiki/File:Spiral_model_(Boehm,_1988).svg

(Boehm & Hansen, 2001)

Key characteristics of bug reports

- **Observed Behavior**: The unexpected behavior witnessed by the user due to the bug. Users should describe what occurred with sufficient detail.
- **Expected Behavior**: The user's anticipation of how the application should behave, contrasted with the actual observed behavior. Users should clarify their expectations, whether it's the absence of an error or the occurrence of a specific error that didn't happen as expected.

- **Steps to Reproduce**: Detailed instructions for the developer to replicate the bug on their system. It's essential to include enough specifics so the issue can be reproduced elsewhere, without necessarily listing the steps in order.
    - **Version**: The specific version of the application being used (Browser) when the bug was encountered.
- **…**

(Davies & Roper, 2014; Zimmermann et al., 2010); online bug-tracking tool: https://www.bugzilla.org/

Exact steps for bug reporting

1. download detailed explanations how to test the different tools from GitHub: https://github.com/FennStatistics/Project_testingCAMtools/tree/main/testing%20materials
2. set up your testing environment
    a. use incognito (private browsing mode), which enables you to browse the internet without your browsing history, cookies, site data, or information entered being saved
3. open respective software (data collection, data analysis, administrative panel) and start testing
    a. links provided in 1.
4. report bugs / possible improvements in the online survey: https://studien.psychologie.uni-freiburg.de/publix/7ENProF1bXN

Hands on bug reporting

Example: data collection (C.A.M.E.L.)

1. download explanations from GitHub:

   https://github.com/FennStatistics/Project_testingCAMtools/tree/main/testing%20materials

2. set up your testing environment
3. open respective software
4. report bugs / possible improvements in the online survey:

   https://studien.psychologie.uni-freiburg.de/publix/7ENProF1bXN

Immerse yourself into exploratory software testing

Online Courses:

- Testing the Burger King website | Exploratory Testing | QA:

  https://www.youtube.com/watch?v=aX42Qr0eeuI

  - see YouTube channel: https://www.youtube.com/@glitchitsystem/videos

picture created by OpenAI, 2024

References

References

- Ambler, S. W. (2005). *A Manager's Introduction to The Rational Unified Process (RUP)*. Ambysoft. https://people.computing.clemson.edu/~johnmc/courses/cpsc372/resources/managersIntroToRUP.pdf
- Boehm, D. B., & Hansen, W. J. (2001). The Spiral Model as a Tool for Evolutionary Acquisition. *Best Practices*.
- Cha, S., Taylor, R. N., & Kang, K. (2019). *Handbook of Software Engineering*. Springer.
- Fenn, J., Helm, J. F., Höfele, P., Kulbe, L., Ernst, A., & Kiesel, A. (2023). Identifying key-psychological factors influencing the acceptance of yet emerging technologies–A multi-method-approach to inform climate policy. *PLOS Climate*, *2*(6), 1–25. https://doi.org/10.1371/journal.pclm.0000207
- Gomes, L. A. F., Torres, R. da S., & Côrtes, M. L. (2019). Bug report severity level prediction in open source software: A survey and research opportunities. *Information and Software Technology*, *115*, 58–78. https://doi.org/10.1016/j.infsof.2019.07.009
- Davies, S., & Roper, M. (2014). What's in a bug report? *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 1–10. https://doi.org/10.1145/2652524.2652541
- Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, *7*(1), 76–80. https://doi.org/10.1109/MIC.2003.1167344
- Munzert, S., Rubba, C., Meißner, P., & Nyhuis, D. (2015). *Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining*. John Wiley & Sons.
- OpenAI ChatGPT. (2024). Knight battling bugs: A motivational illustration for software testers [Digital image]. Created March 19, 2024.
- Winters, T., Manshreck, T., & Wright, H. (2020). *Software Engineering at Google: Lessons Learned from Programming Over Time*. O'Reilly Media, Inc.
- Zimmermann, T., Premraj, R., Bettenburg, N., Just, S., Schroter, A., & Weiss, C. (2010). What Makes a Good Bug Report? *IEEE Transactions on Software Engineering*, *36*(5), 618–643. https://doi.org/10.1109/TSE.2010.63