

SOFTWARE PROJECT QUALITY AWARD



accenture

WHO ARE WE?

ACCENTURE

- Accenture has a 20+ year long history with the Software Project course, coaching and evaluating project teams on achieving high quality project deliveries
 - QA prize will be once again handed out to the best project delivery!
- Accenture team for this semester:
 - Niina Jones, Jarno Hilvenius and other guest experts
 - You can reach us by email:
Firstname.lastname@accenture.com
- Accenture will host 2 EES sessions during the course
- Accenture will also arrange the **Software Project and Quality Award Gala in the end of the course**



OUR BUSINESSES

OPERATING GROUPS



COMMUNICATIONS,
MEDIA & TECHNOLOGY



FINANCIAL
SERVICES



HEALTH &
PUBLIC SERVICE



PRODUCTS



RESOURCES

GROWTH PLATFORMS



accenturestrategy

accenturedigital

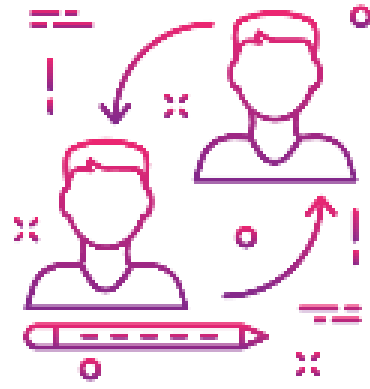
accenturetechnology

accentureoperations

WORKING AT ACCENTURE



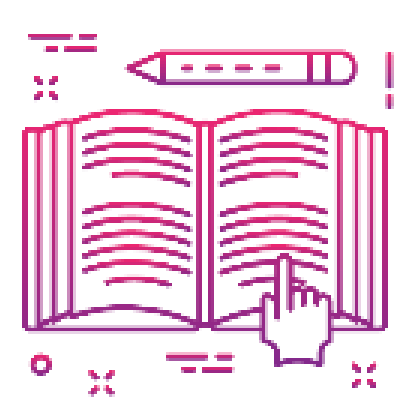
Flexible work



**Possibility to change
your career path**



**Great colleagues
and community**



**Training
and learning**

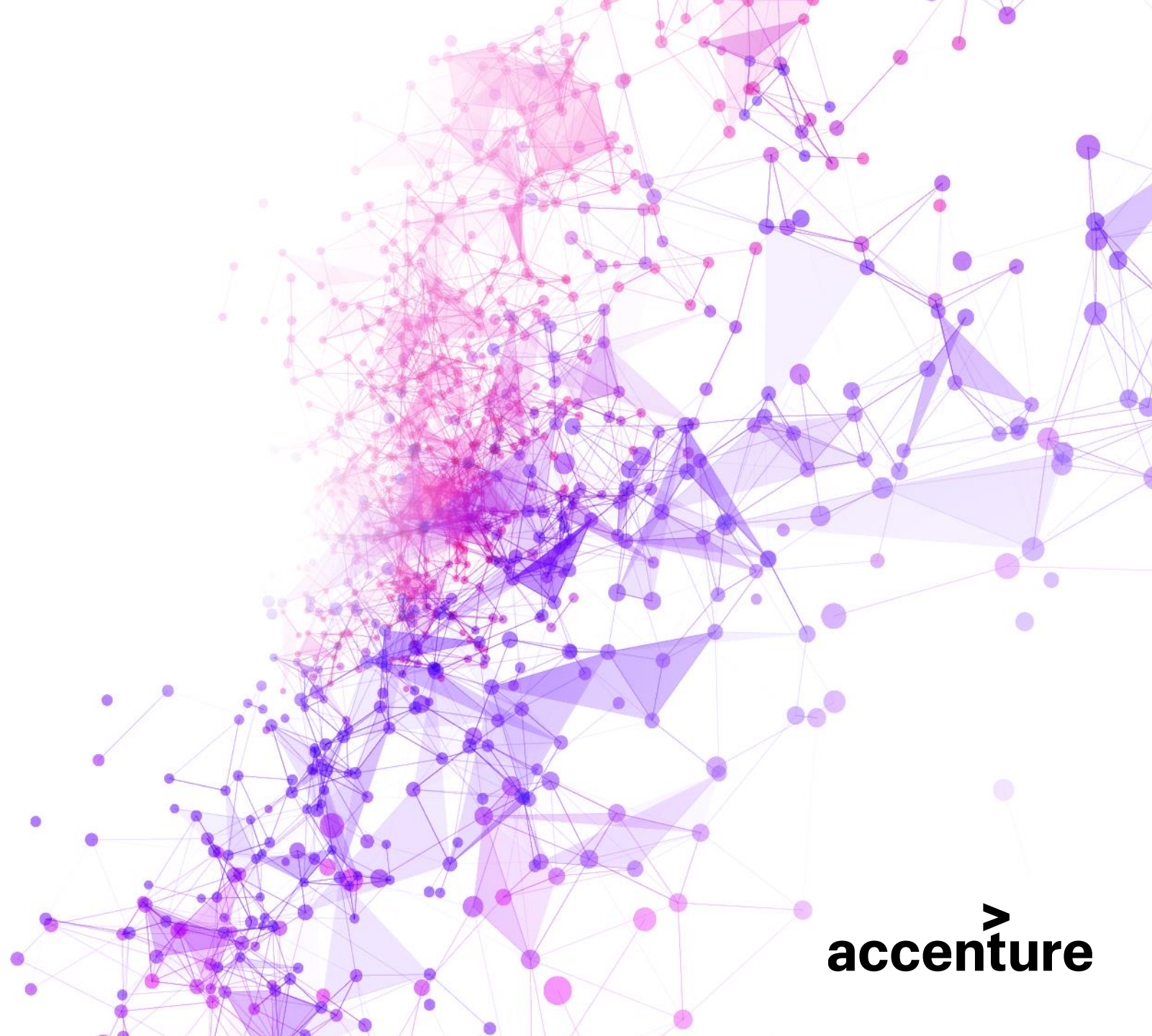
WHAT DOES IT MEAN IN PRACTICE?



CLOUD
FRONT END
BACK END
ARCHITECTURE
SYSTEM DESIGN
AI & MACHINE LEARNING
RAPID PROTOTYPING
AGILE DELIVERY
UX AND SERVICE DESIGN
AND THEN SOME...



**AND THEN THERE IS
THE PRIZE FOR THE
BEST PROJECT ...**



accenture



**AND THEN THERE IS THE
PRIZE FOR THE BEST
PROJECT...**

ACCENTURE QA AWARD

In the end, selecting the winner is based on the impression of quality of your work and project delivery

[Course points, client satisfaction, technical quality of the work, working methods, solution extendability, project management ...]

Winner will be announced and receive the award in the end of the course

Feel free to approach the Accenture team in case you have any questions related to project delivery work in practice

We are committed to arranging one support session per project, on any topic, should you need support on some topic during your projects

The Accenture logo, featuring the word "accenture" in a bold, black, sans-serif font. A yellow chevron symbol is positioned above the letter 'u'. The logo is centered within a white circular area that is part of a larger grey graphic on the right side of the slide.

PRIZE
1 200 €

SO WHAT IS **QUALITY**
IN SOFTWARE DESIGN?

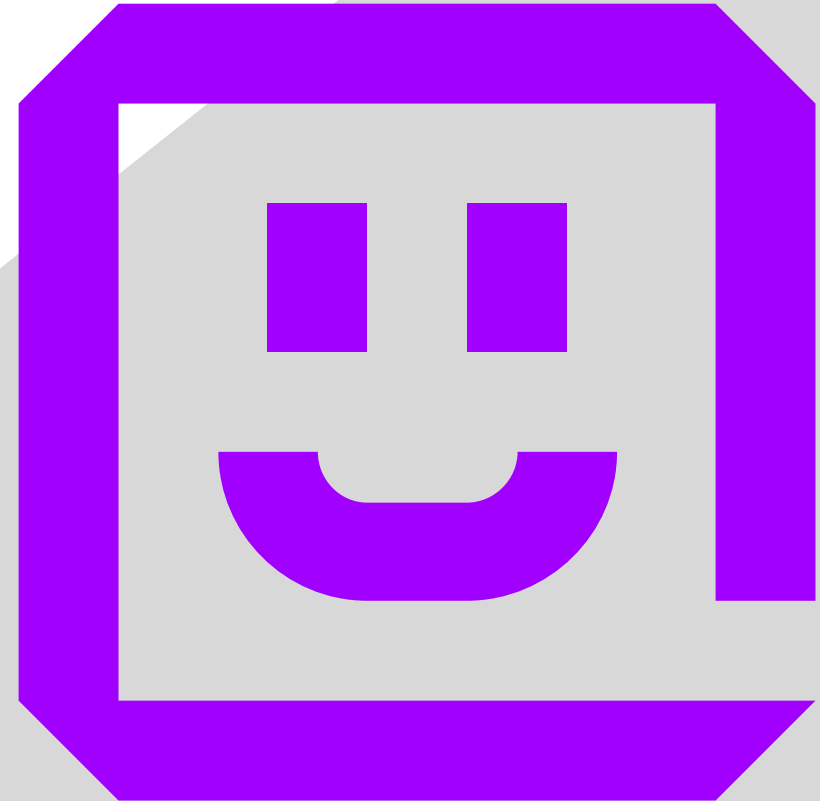
**“ALL CODE IS GUILTY
UNTIL PROVEN
INNOCENT.”**

- UNKNOWN GREAT THINKER ON THE INTERNET

QUALITY CRITERIA

Customer satisfaction and value

- Points given by the customer
- What happens to the end product after the project ends



QUALITY CRITERIA

Quantifying the quality

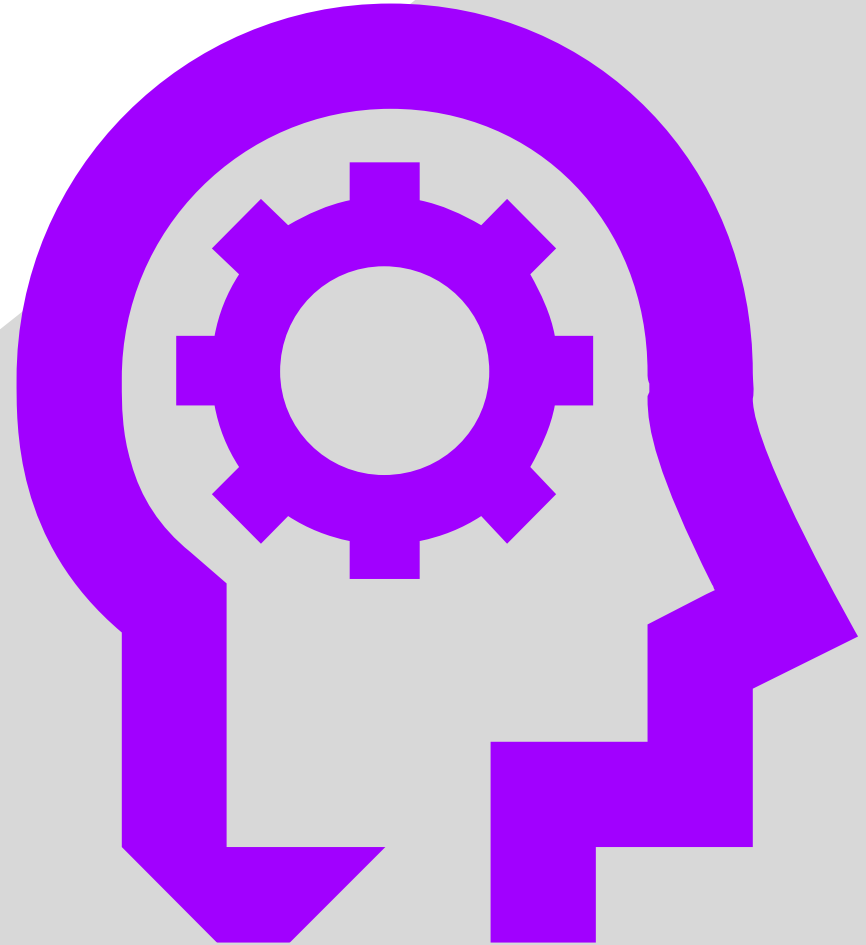
- **Technical quality of the code itself, and how you quantify it**
- **Functional quality, i.e. the end product and functionality**
- **Quality of documentation**



QUALITY CRITERIA

The degree of difficulty

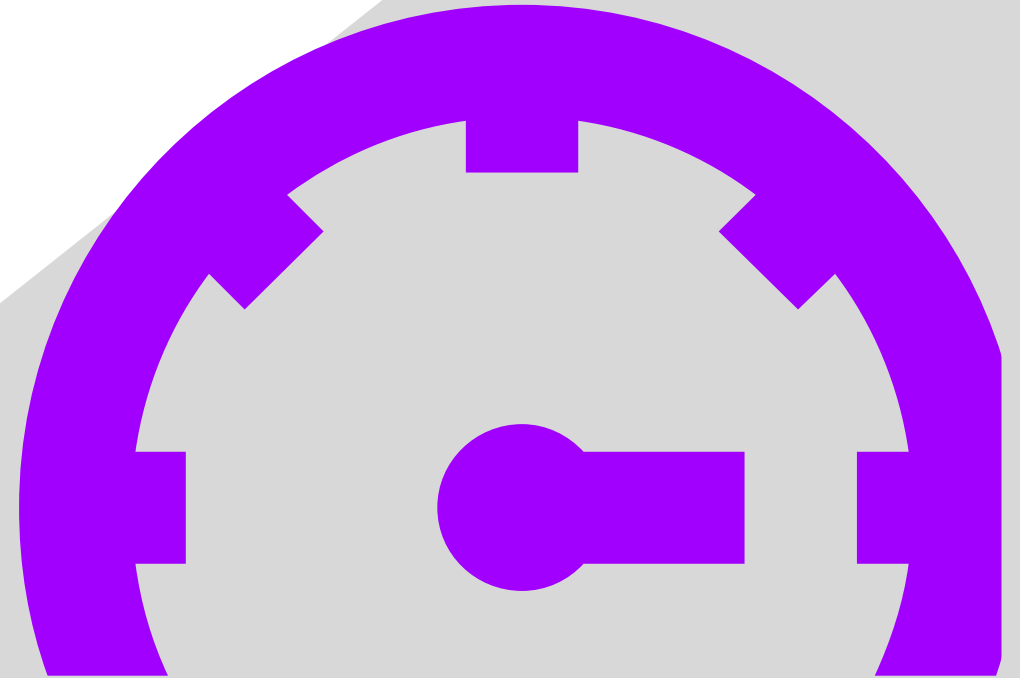
- The scope and difficulty of the project
- The amount of end products and amount of functionalities you have delivered
- The degree of customer's involvement and how successfully has the project end hand-over been executed



QUALITY CRITERIA

Work efficiency

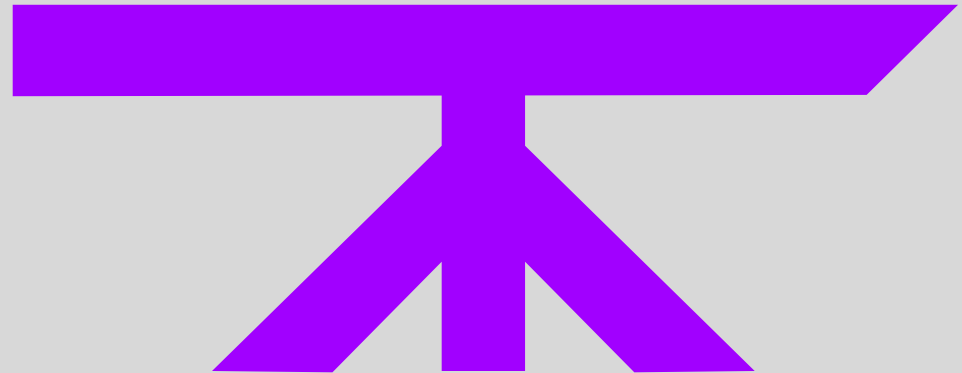
- Are you utilizing the whole team effectively
- Tools and methodologies being used and how effectively
- Effective use of open source, already available solutions etc.



QUALITY CRITERIA

Demo

- You have a clear, well prepared and presented demo
- Well built story around the end product



TIPS FOR ACHIEVING HIGH QUALITY 1/3

Extendability

- Closely collaborate with the client during the technical development to ease the handover at project end
- Well documented system, and tools used to build the system also help
- Strive for well structured high quality source code, with thorough comments

Responsibilities

- Share and assign responsibility in all the critical areas (architecture, quality assurance, project management etc.)
- Scrum masters are expected to create and ensure productive working environment for the whole team
- This is a team effort, so commit to the work → Every team member achieves the expected amount of hours for the work

External stakeholders, including the client, always bring risks to the project

- Consider e.g. software, hardware, and licenses related to your work products

Opt for free-of-cost, well-known open source software where possible

- These are usually immediately available, well tested, and discussions and instructions are available online

TIPS FOR ACHIEVING HIGH QUALITY 2/3

Automate

- E.g. development process and reporting automation, code review automation, testing automation, build and deployment automation...

Agree on the internal processes of the group and adhere to them

- Visualize the most important/complex processes e.g. as decision swimlanes and make them available for everyone e.g. on a project Wiki page

Benchmark against data from previous years

- Where has the time been spent, what kind of project has it been, and what has been the end result?

Pay attention to clear reporting, e.g. regarding your result and use of time

- Track your efforts closely so you can analyze the data and identify opportunity for improvement
- This is one of the most important factors when selling / delivering projects in the real world
- It is also one of the hardest thing to get right for many teams

Carefully allocate your use of time

- E.g. arrange meetings with just the required people → save efforts and frustration from others → budget you can allocate elsewhere

TIPS FOR ACHIEVING HIGH QUALITY 3/3

Communication: for speed and comprehension, the following applies

- Face-to-face > video conferencing > phone call > instant messaging > email > regular mail > messages in a bottle > assuming that the teammate / client / any counterpart already knows and understands

Take full use of external testers

- Outsider can take a fresh / new perspective
- Closely limit the scope of testing to get the most relevant results

For reporting, consider how, what, and most of all WHY

- Course requirements must be met even in case of simplifying the approach in some regard
- A picture is worth a thousand words, e.g. when presenting the technical architecture
- E.g. the commonly used LOC metric (Lines Of Code), what is the targeted message?
 - High number = high amount of poor quality code and few functions, or
 - Low number = little amount of high quality code and many functions
 - In extreme cases, code refactoring can reduce the LOC → does the plain LOC metric say the work been counterproductive?

Put effort in preparing your project demos!

- Practice in advance (demo dry-runs)



**THANK YOU
AND GOOD
LUCK!**

accenture