

Project Presentation Guidelines

2017/2018

Pedro Cabalar

Dept. Computación
Universidade da Coruña

26/04/2018

What a project should be

- Your project is a **very important part of your training**
- An **opportunity to show** your abilities and capabilities
- A good project is your **best business card**. Even better than good grades
- **However**, if a good work is **badly presented**, it is as if it **never existed!**

What a project should be

- RD 1393/2007 (Bolonia) says:
“the graduate project is done in the final stages of an undergraduate programme and should be used to **assess the career abilities**”
- Regulations for Computer Science, Corunna (FIC) can be found at <https://wiki.fic.udc.es/docencia:grao:tfg:indice>

What a project should be

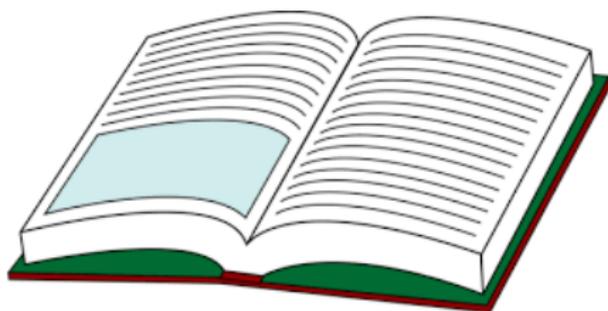
FIC regulations say that a project:

- should contain **original, individual** work about Computing Engineerings (Ingeniería Informática)
- must belong to **one career specialisation** from the five:
 - 1 Software Engineering
 - 2 Computer Science
 - 3 Information and Communications Technologies
 - 4 Computer Engineering
 - 5 Information Systems

What a project should be

- FIC regulations also talk about three types of projects:
 - ① **Classical Engineering**: building systems, devices or processes related to Computing Engineerings
 - ② **Development for Research**: systems for research support.
 - ③ **Standards Assessment** or **Methodological Nature**.
- Your project **specialisation and type must be clearly stated** both in the report documentation and in the project defence.

The report



The report

- In the FIC's URL you will find some instructions regarding the **document formats and structure**:
 - A short guideline about the document format
`https://wiki.fic.udc.es/_media/docencia:grao:tfq:directricestfg_memoria.pdf`
 - Instructions for electronic documents or contents (CD/DVD)
`https://wiki.fic.udc.es/_media/docencia:grao:tfq:directricestfg_cd.pdf`
 - Illustrative guide for report structure and contents
`https://wiki.fic.udc.es/_media/docencia:pfc:estructura-memoria-pfc.pdf`
 - Illustrative guide for user and reference manuals
`https://wiki.fic.udc.es/_media/docencia:pfc:estructura-manuales-usuario-y-referencia-pfc.pdf`
- Apart from the format, let's see some general recommendations

...

Document structure

- Summary and keywords
- Introduction
- Background
- Main work
- Evaluation
- Related work
- Conclusions
- Appendices
- Bibliography

The summary

- First page, but **last thing you write!**
- It should quickly answer the question:
What's this project about?
- Short (1/2 page at most) and **focused on the work** you did, results or contributions
- We don't make any introduction. Directly **begin with** "In this project . . ." or "This project consists in . . ."
- In the summary, all **acronyms must be defined** and clarified. Avoid bibliographical citations in the summary

Keywords

- Relevant nouns or noun phrases. Example:
agile methodologies, extreme programming, java spring, online teaching
- Original purpose: **facilitate bibliographical search**
- Nowadays web **search tools** (Google, Yahoo, etc) make a better job. They can decide the statistically relevant words
- Recommendation: maintain an **uploaded PDF version** of your project documentation

The Introduction

- The **introduction is the most important part** of the report. It should **catch the reader's attention**. Structure:
 - A general **context**. Example: software in schools
 - A **problem to solve**. Example: systems for online exams
 - **Existing solutions**, their advantages and problems
 - The project **proposal**: an online test tool for primary school grading
 - **Very important: the main goals** (better an enumeration).
Be **specific**: avoid things like “improving the teaching quality”
Never: “the goal is to fulfill my degree in Computer Science”
Sometimes, you can also specify **what won't be covered**
 - The **structure** of the rest of the document:
“Chapter 2 describes. . . , Chapter 3 . . . ” etc.

The Background Section

- Include here the **relevant fundamental knowledge required**: technologies, methodologies, mathematical definitions, etc.
- Choose a **good selection criterion**:
 - Example 1: if you use a rare programming language or unusual methodology, give some hints of its general behaviour and include more details in an Appendix but do not include a C or Java manual!
 - Example 2: most people will know about HTML, but not everybody will know the new features of HTML5

The Background Section

- The background must describe concepts, tools or methods **created by someone else, not you!**
These things cannot be in *your* “Main work”
- Usually, you go filling the background section depending on **what you need** for writing the “Main work”
- **Relevance** principle: do not include background content if it is not used afterwards

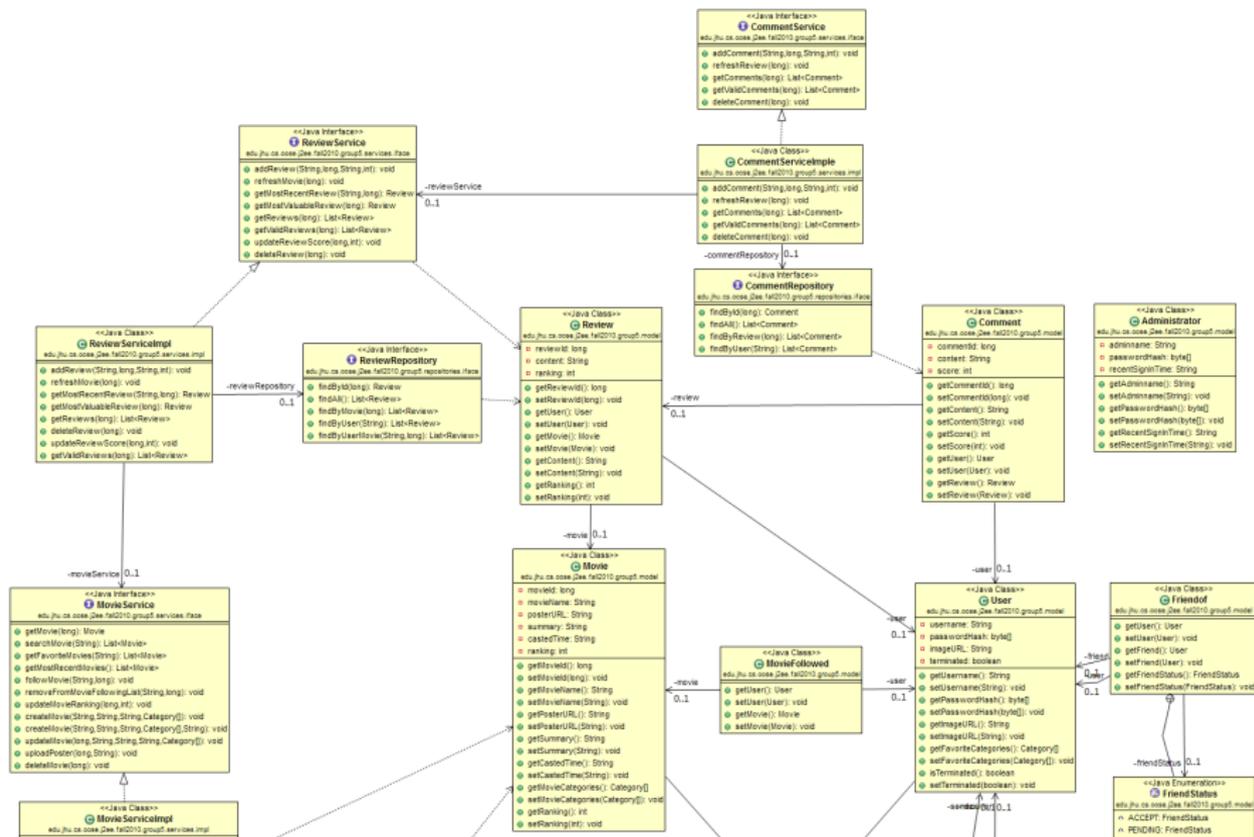
Main Work

- It describes your **main results and contributions**
- You can structure this part into **as many sections as needed**.
- All engineering projects must contain a **project planning** with an economic budget.
- **Common mistake**: lots of analysis documentation, design diagrams, even code, but no text explanations or structure.
- Typical example: complete set of use cases, class diagrams, sequence diagrams, etc, etc. This is definitely **wrong!**
- The “Main Work” section must keep being a **readable textual description**

Main Work

- Use an **Appendix** for all the development documentation (analysis, design, code). You can refer to it in the text if needed.
- In the “Main Work” text you can include diagrams, but when they contain **salient, relevant information**
- These diagrams can be **simplified**, removing information that is not relevant for the explanation.
- And remember: an **unreadable font** is unreadable by definition

What would you expect from this?



Main work

- This kind of diagram should have full size and (double page) and be included in the Appendix
- In the “Main Work” just **leave the class names** and perhaps some relevant methods you want to emphasize
- A similar criterion applies to **code, database description**, etc. Include only relevant pieces.

Evaluation

- Depending on the type of project you may have to include an evaluation section.
- If you doubt whether it is necessary, then it is necessary.
- This section answers the questions: **how good** is my system or approach? **how does my system perform** on different scenarios? **which properties** can be guaranteed? ...
- Example: the online test tool. Try the tool on different platforms. Test performance for different number of users.
- **Evaluation measures**: execution time, memory usage, error margins, etc.

Evaluation

- For some types of tools, there exist **available benchmarks**. If so, use them. Example: corpus texts for information retrieval, problems for AI planning, etc.
- Sometimes, a project has been **implemented in a real scenario** or with real users. This is a **very valuable feature**. Explain and emphasize this situation.

Comparison to Related work

- As with evaluation, the inclusion of this section may **depend on the type of project**. Sometimes, **evaluation and comparison can be joined**.
- **Compare to other existing approaches** for a similar goal. If fundamental knowledge is required, include it in the Background section.
- It may suffice with mentioning the **main differences** or it can be a very detailed comparison.
- Example: show the features offered by other online test tools, and compare the advantages and disadvantages (functionality, performance, code availability, better design or implementation, etc) to your system.

Conclusions

- The conclusions have three parts:
 - A **summary** of the work you completed
 - The **conclusions** you drew from the project
 - The **open topics and future work**
- The **summary** can be done, grossly speaking, changing the verb tenses from the goals in the introduction:
 - Introduction: “We **will develop** a tool that **will be** used . . .”
 - Conclusion: “We **have developed** a tool that **has been** used . . .”

Conclusions

- Explain the **obtained conclusions**. Be **honest, critical and constructive**.
- The conclusions will contain **both**:
 - **Positive**: advantages of the system, lessons learnt, etc
 - **Negative**: difficulties found, incomplete aspects, weaker parts, etc
- **Future work**: explain those aspects in which the project could be extended in the future

Appendices

- Include in the Appendices any **detailed** documentation you consider convenient but that would perhaps break the readability in the main text.
- Example: full set of UML diagrams, datasets used for tests, etc.
- The document should be as much **self-contained** as possible (for a Computer Science reader).

Bibliography

- The bibliography should follow IEEE guidelines:
<https://iee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf>
- All bibliographical **citations must be referred** in the text. Don't include entries that are never cited! **Background section** should include most of the citations.
- Cite the **original source**: avoid blogs, wikipedia entries, generic handbooks, etc.
- You can also include a (word) index, lists of tables and figures and, **very important**, a **list of acronyms**.

Writing style

- Avoid using singular first person

“I have used Hibernate because **my** experience . . . ”

“**We** have used Hibernate because **our** experience . . . ”

“Se ha usado Hibernate porque la experiencia previa . . . ”

- Make a **reasonable, coherent use** of the language. If you choose Spanish or Galician, check the existence of suitable translations for technical terms

“Definimos varios **templates** de **screens**”

“Definimos varias **plantillas** de **pantallas**”

- Exception: some words have become part of the *de facto* **jargon** without a good, accurate translation: buffer, device driver, shell, script

Writing style

- Spanish/Galician: **don't use infinitives to start a sentence**
“Comentar que este aspecto ha resultado laborioso”
“Cabe destacar que este aspecto ha resultado laborioso”
“Como comentario, este aspecto ha resultado laborioso”
- **Don't abuse acronyms.** Keep the document readable to other students or Computer Scientists from different specialisations. Remember to include an acronym list, anyway.
- Spell checkers are always required, but they are not enough.
Proof read your document!
“A parte de esto, se a echo un modelado de . . .”

Page headings

- Page headings are used for a **quick browsing** of the physical volume
- Include **chapter title on the left** and **section title on the right**
- Don't include the project title or your name: these are constant for the whole document!
- If you start a **chapter, do it on the right**, leaving a blank page on the left.

Text editing

- If you need many formulas, use \LaTeX

$$h^-(X|Y) \leq \frac{n+1}{e} - h(X|Y) + \int p(y) \log \left(\frac{\mathbb{E}[|X|^2 | Y=y]}{n} \right) dy \quad (1)$$

```

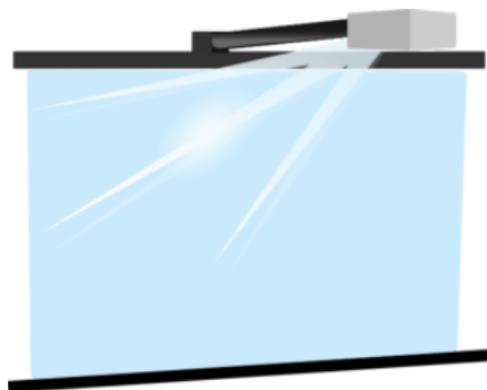
\begin{multline}
h^{\{-}(X|Y) \leq \frac{n+1}{e} - h(X|Y) \quad \\\
+ \int p(y) \log \left( \frac{\mathbb{E}[|X|^2 | Y=y]}{n} \right) dy \quad \\\
\end{multline}

```

Text editing

- If you don't need formulas, consider \LaTeX anyway!
- More learning effort but will save a lot time in low-level formatting afterwards

The slides



The slides

- The slides are your **main support** for the presentation
- They should be **smooth**. Adopt a **minimalistic style**. Avoid decorations: they easily become distractions.
- The slides **don't need** to follow the report structure!
- In the report: you give precise definitions and detailed descriptions
In the slides: you explain the ideas **using examples** instead

Cover

- The first slide is a “cover” like:



A web tool for online tests Mickey Mouse

Director: Walt Disney

Grado en Ingeniería Informática
Mención en Ingeniería del Software

Proyecto clásico de Ingeniería
Facultad de Informática

A Coruña, 7 de abril de 2014

Motivation

- After the cover, we **jump to motivation**. Don't include an outline yet.
- As in the report, introduce the context and the problem to solve.
- The introduction is crucial: it must **catch the audience's attention**
- Use a realistic **running example**, a practical case or scenario.
- After the motivation, you show the **outline** of the rest of the presentation.

Demo soon

My recommendation: If you plan to use a **demo, this is the best moment.** Why?

- The audience is more **receptive** now
- You illustrate soon and **exactly what you did**, you will explain **how** later
- If your presentation is bad, it will happen later: many projects begin with a **bad presentation**, are not well understood and we end up **without time for demo**
- Remember: we are Computer Scientists. What we better estimate is **working software!**

But demos are tricky!

Murphy's laws imply that the **demo will fail**

- 1 Make an **essay with the same equipment** (laptop, network connection, video projector, etc) the day before.
- 2 Prepare a **B plan**: if the live demo fails, use a **pre-recorded video** with the essential execution steps. Note: if you use video from the beginning **there must be a justification**
- 3 And a **C plan**: if the video doesn't work, use a set of **pre-organised screenshots**

Remember: the demo is **very valuable**, but if it fails, it will only make you **waste a lot of your presentation time!**

Main work

- In the slides you don't need to separate background from main work
- Make a **careful selection** of what you need to tell
- You will be able to read around **1 slide per minute** (without counting outline slides or transitions). Make a reasonable use of pauses and animations
- Use short sentences. **Emphasize keywords** in each item
- Avoid small fonts or long paragraphs.

Main work

- **Common mistake**: reusing text from the report. Example:
- After analysing different alternatives, we have decided to follow an agile methodology, especially due to the high uncertainty in the set of requirements. Among the different agile methodologies, in particular, we have chosen Extreme Programming for its simplicity and because it is recommendable for small project sizes

Replace by

- **Extreme programming**: agile method, better for small projects

Evaluation, related work and conclusions

- These sections are usually **less important**. If run out of time, you can skip them
- Related work, for instance, can be omitted in the presentation
- For **conclusions**, follow the same three parts from the report
 - 1 Summary
 - 2 Positive/negative conclusions
 - 3 Future work

The last slide

Jump to a modified cover:



A web tool for online tests

Mickey Mouse

Director: Walt Disney

Thank you for your attention!

Grado en Ingeniería Informática
Mención en Ingeniería del Software

Facultad de Informática

A Coruña, 7 de abril de 2014

The presentation



The presentation

- Try to **convince the audience** and communicate your work
- **[My opinion]** Warning: you are **not a commercial**. Remember you are presenting an **academic work**
- **Don't read any prepared text** by any means. The audience will immediately disconnect!
- Use your own words. Use your **voice to emphasize** key concepts. Avoid a monotone voice tone.
- People in the audience **get tired** if they have no **idea of progress**. Show the **current slide number** and the number of slides
- The slide number is **crucial to facilitate questions** by the committee members.

The presentation

- **Practice** your presentation several times. Use some friends or colleagues as improvised audience. Tell them to make questions.



- Time control: keep **a clock or timer at sight**.
- ... If you **suffer a delay**, you may skip evaluation, comparison and conclusions, but don't try to read all the slides in a hurry!



The presentation



- If at any moment you get **nervous** \Rightarrow **focus on the details** of your work. No one like you will know what you made
- ...
- **Golden rule:** if you **go blank** \Rightarrow start **reading the slide** for a while ... and you'll soon recover the flow

Minor protocol guidelines

Main rule: Never do anything until **the committee tell you so**

- Start the presentation after the committee chair's introduction. Read the project title and your name **only if the chair didn't do so**
- You may make rhetorical questions to the audience or committee, but **never force them to answer**. You are not teaching in a classroom!
- Don't end the presentation with:
"Now you can make any questions"
Say instead
"That's all. Than you for your attention"
and wait for the chair's observations or questions.
- **Don't interrupt** their questions or observations. Remember: the committee members can be informal but not you.

Project Presentation Guidelines

2017/2018

Pedro Cabalar

Thanks for your attention!

Facultade de Informática
A Coruña, 26/04/2018