**General Guide for TU Delft TI3800 Computer Science Bachelor Project (2014)**

The TU Delft Computer Science Bachelor Project (also referred to as "*bachelor eindproject*" or "BEP") is the final project carried out by Computer Science bachelor students and the last course of the undergraduate Computer Science curriculum. The bachelor's project is carried out in a group of 2-4 students, who work together as a software development team. During the project, the team produces a product that is commissioned by a Client ("opdrachtgever") and solves a real-world problem. The Bachelor Project also puts special emphasis on research skills: the students are expected to carry out research and demonstrate that they have arrived at optimal solutions, taking the full range of possible solutions into account. Over the course of the project, the student team experiences the full trajectory of a real-world software development project, including:

- research and problem analysis,
- requirements,
- specifications and quality requirements,
- implementation,
- testing and validation,
- delivery of a working product, including necessary documentation.

The team tackles this challenge by integrating their knowledge and experience from previous courses and projects in the curriculum.

**Purpose of this guide** This guide is written for both the student teams carrying out the bachelor project and also the person who coaches the team from the side of the EEMCS (EWI) faculty at the TU Delft (the "EWI Coach" or the "TU begeleider"). The guide summarizes the requirements of the bachelor project and how the project is evaluated. It also gives practical that will help students and coaches to avoid common mistakes. The guide is *not* a set of directions for how to solve the problem represented by the Client's commission. The real-world nature of the bachelor project means that each time students must determine their own way of approaching the problem and their own solution.

**Summary** The following is handy summary of the requirements of the bachelor's project:

**Ten important numbers for the Bachelor's Project**

1. 2-4 students per team.
2. 2 advisers (1 Client adviser and 1 TU Coach from EEMCS).
3. 1 project plan between TU Coach and team (*plan van aanpak*).
4. 2-week research phase
5. 10 page research report (from research phase).
6. 2 submissions of code to the Software Improvement Group (SIG) for evaluation.
7. 10-11 total weeks of work (420 hours).
8. 30-50 page final report (delivered to the committee 7 days before the presentation)
9. 3 Bachelor's Project committee members (1 TU Coach, 1 Client, and 1 Bachelor Project Coordinator)
10. 30-35 minute final presentation (including demo).

**Prerequisites for the Bachelor Project:** Students will be admitted to TI3800 only if they have finished both the "Propedeuse" exam and all second year course. In case of special circumstances, students should submit a ""Verzoek toelating vak"[1] (request for admission to a course) and, as a second step, if necessary, discuss the situation directly with the study advisor.

**Bachelor Project Coordinators:** The Bachelor Project Coordinators for the academic year 2013-2014 are: Martha Larson (m.a.larson@tudelft.nl) and Felienne Hermans (F.F.J.Hermans@tudelft.nl) Please contact them if you have questions after having read this guide.

**Role of the Client adviser:** The Client is the real-world stakeholder, who commissions the team to develop a software solution that addresses a specific problem. The Client is usually a company, but can also be an entity from within the TU Delft. The problem posed by the Client should be "open" in the sense that no formal software specifications exist for the system commissioned by the client. Instead, the team, during the research phase, must themselves determine the optimal solution, by gathering and comparatively analyzing the necessary background information. More information for clients is available in the "Guide to Proposing a TU Delft Computer Science Bachelor Project"

http://homepage.tudelft.nl/q22t4/Resources/ProposersGuideTUDelftCSBachelorProject.pdf

**Role of the TU Coach:** The role of the TU Coach (EWI Coach) is to represent the educational interests of the TU Delft. S/he acts as a guardian of the learning objectives of the bachelor's project. The primarily function of this role is guiding the bachelor students in applying their skills and knowledge within the bachelor's project. The TU Coach should support the team in choosing the appropriate software development methodology. Practically, the TU Coach should offer support in keeping the bachelor project running along its timeline and converging to a timely, finished project. Typically activities of the Coach are: Approving the project plan, meeting regularly with the team (usually once a week), giving any needed feedback during the preparation of the final report and the final presentation.

It should be noted that the Coach provides a counterweight to the interests of the client, whose goal is a working piece of software that solves a particular problem. The TU Coach should ensure that the students understand the importance of the research aspect of the bachelor's project, which is part of the educational interest of the TU Delft, but may not necessarily be the interest of the client. For more information on the Learning Objectives of the Bachelor Project see "Appendix 1: Learning Objectives of the Computer Science Bachelor Project" below.

**Role of the Bachelor Project Coordinator:** The Bachelor Project Coordinator checks the project proposal made by the client. Once the team has identified a TU Coach, the Bachelor Project Coordinator and the Coach together gives the team the green light to begin the project. In general the team will not see the Bachelor Project Coordinator again until the final presentation. However, in case of specific issues, the team should not delay in contacting the Bachelor Project Coordinator.

**Finding a project and a team**: In order to find a project, students should create an account in BEPSys http://bepsys.herokuapp.com the TU Delft Computer Science Bachelor Project System. Here, students can browse potential projects, and/or place an "advertisement" stating that they are looking for a team with which to carry out a project. Students who are interested in an open project can contact the company that has proposed the project for more information. Also, if there are already students registered for the project (visible in BEPSys), please contact those students to ask about joining their team.

In order to get a project started a series of steps must be carried out:

[1] See: http://studenten.tudelft.nl/nl/ewi/formulieren/bsc-opleidingen-ewi/

- Students determine whether they fulfill the requirements to carry out the Bachelor Project. Students who need to apply for an exception, should take action as soon as possible. One person in the team who doesn't fulfill the requirements can hold up the whole team.
- A team of 2-4 students has joined an "open project" in the BEPSys.
- The study advisor formall approves the request of each member to start the Bachelor Project (The BEPSys provides support for the approval process).
- The team sends and "official" invitation to their TU Coach (a.k.a. EWI begeleider). If a project description already lists a TU Coach, then the Coach has been designated in advance. The team should contact that person informally to get their verbal approval before triggering the system to issue the official invitation. If the project description does not list a TU Coach, then the team should approach members of either the Intelligent Systems Department or the Software and Computer Technology Department and invite them to coach the project. Again, it is important to have a discussion with the person outside of the system, before issuing the official invitation. By accepting the official invitation, the TU Coach is signaling his/her approval of the project.
- The project must pass the Bachelor Project Coordinator for a final approval and the bachelor project team is clear to start.

It is also possible for students to find their own project opportunity. In such a case, students have contact with a client and ask the client to submit a proposal for a project to the BEPSys. The proposal should contain the names of the students in the "Additional Information" field so that it is clear that this project is specifically for these students. When communicating with companies, the students should provide them with the link to the "Guide to Proposing a TU Delft Computer Science Bachelor Project"

http://homepage.tudelft.nl/q22t4/Resources/ProposersGuideTUDelftCSBachelorProject.pdf

in order that the company understands the characteristics of a TU Delft Computer Science Bachelor Project. In case of uncertainty, please contact the Bachelor Project Coordinators.

**Bachelor's project timeline:** The team is expected to spend 420 hours (per person) working on the project over the course of 10-11 weeks. The students usually carry out the project in the fourth and final quarter of their bachelor's course of study, but this is not always the case. If the students are carrying out the bachelor's project on a part-time basis, they should be devoting at least 24 hours a week to it. The TU Coach should meet with the students on a need-to-meet basis. In practice, the frequency is approximately once a week and in general should not be less than once every two weeks. The students should prepare these meetings and send the documents to be discussed ahead of time. During these meetings the TU Coach approves the documents. The key documents are: the project plan (which should be finished as soon as possible at the beginning of the project) the research report (which should be finished after two weeks) and the final report (which should be finished so that it can be sent to all members of the bachelor project committee 7 days before the final presentation).

**Relating to the client companies:** The bachelor project is not an "internship". In other words, the students are *not* simply company employers for two and half months. Rather, in this time they are expected to carry out a full software development project complete with a research component. The TU Coach does not have to necessarily visit the company, but if the opportunity exists to visit or to hold joint meetings with the client there, it can be helpful for the project.

**Contractual/legal aspects:** Students must set up their own agreement with the Client. Students enter into this agreement as their own entity (i.e., they are *not* part of the TU Delft). The agreement between the students and a company acting as a Client does *not* involve the TU Delft. It is important that the students take responsibility themselves for any agreement that they make with the

company. Some companies will wish to enter into a formal contract with the members of the team. Such contracts cover aspects such as insurance, intellectual property and non-disclosure. In terms of intellectual property, if no agreement is made the "*opdracht verzinner*" (i.e., the person who formulated the project, usually the Client) retains the rights to the intellectual property. The TU Delft does not officially support students in making the decision about what kind of contractual agreement they would like to enter into with the company. However, we will make an example contract available for students to compare with contracts used by companies. We will also make an effort provide some general guidance about contracts in case there is a point of great uncertainty. The ultimate responsibility for the decision on the contractual agreement with the Client lies, however, with the student.

It is important to keep in mind that the final report of the bachelor's project is a public document and that students cannot carry out a project whose results the company is not willing to publicly disclose. In some cases, students are able to put proprietary information into an appendix. This appendix is then removed before the report is uploaded to the repository. Students should avoid projects involving proprietary information that is not separable from the main body of the report in this way.

**Conflict resolution:** Conflict can be avoided if the team members state their expectations of each other and of the team in an explicit and clearly formulated manner at the start of the project. For example, "If there is a danger that a deadline cannot be met, a team member must inform the rest of the team 24 hours in advance." If a disagreement arises in the team or with the Client, it is important to take action immediately so the project is not endangered. The team should schedule a resolution meeting with the TU Coach. The TU Coach will ask which agreements or expectations of the team have been violated and will help the team lay out a plan for resolving the disagreement to the mutual satisfaction of the members. If a resolution meeting with the TU Coach is not an option, the team should contact the Bachelor Project coordinators.

**SIG** The students are required to send the code that they produce two times during the course of the project to the Software Improvement Group (SIG), once at the 75% mark at the latest and once at the end. In the regular situation of 11 weeks for the project, this would be in the 8th week. SIG sends them a score and feedback. For projects running "on schedule" in Q4 of the academic year 2013-2014, the software should be submitted to the SIG on 13 June 2014. The feedback from the first code submission should be integrated into the code for the second code submission. The final report includes discussion about the SIG feedback and what the student team has done to address it. Questions about the SIG code submission should be directed to Eric Bouwers (e.bouwers@sig.eu).

**Project plan:** The project team should create a project plan at the beginning of the project (within the first few days is advisable) together with the Coach and the Client. The plan should be approved by the project Coach. Teams should make continuous reference back to the plan during the course of their project in order to confirm that they are remaining on track.

**Agreements with the Coach and Client** It is important to note that the description of the project in the BEPSys is not binding. For this reason, it is important that all agreements with the Client and the Coach concerning the content, goals and methods of the project be set out in writing. The project plan can be used for this purpose. If anything else changes over the course of the project, however, the change should be documented in written form. Often an email describing the change is sufficient for this purpose. Students should not rely on verbal agreements with the Client or Coach for the important points.

**Research report:** The research report summarizes the results of the research phase of the bachelor project. It should convince the reader that the team has chosen an optimal approach to their problem. Specifically, the relevant alternative solutions should be investigated and compared.

All aspects, e.g., choice of algorithms, choice of frameworks, choice of component technology, should be well motivated. The report should include the key references that support the team's choices. For more information see "Appendix 2: Guidelines for the Final Report" below.

**Final Report** There is no pre-set or required format for the final report. The report should document that the project was a real-world software development project and demonstrate that the students have researched their approach (they should incorporate the original research report into the text of the final report). It is important that the report reflect that the students have applied what they have learned during their bachelor's studies and, in particular, that at the end of the project that they have demonstrated that the resulting software provides a solution to the problem defined at the beginning of the project (they should make sure to plan enough time in their project to carry out evaluation and to critically reflect on the process and the product that they have produce.) The report should allow a new team with a general technical background to understand what was done in the project and why. After reading the report, a new team should have enough experience to be able to use or build upon the output of the project. Previous reports are in the repository (http:// repository.tudelft.nl) "faculty:"Electrical Engineering, Mathematics and Computer Science" type:"Bachelor thesis".

**Final presentation:** The final presentation should be aimed at audience with a general technical background, but who do not have detailed prior knowledge of the bachelor project. The presentation should present the whole project, from motivation through to evaluation. It should demonstrate to the audience that the team has achieved the goal of the project. It should also convince them that the team possesses the software development skills to be equally successful in another project. The final presentation should be about 35 minutes long including the demo. The team should be prepared to answer questions on their project after the presentation. In preparing their final presentation, students should consider the basic elements of a good presentation. Review the principles of presentation learned during the Bachelor Program. It is very important to practice the presentation.

The student team that they need to make their own arrangements for their final presentations: reserving a room and setting up a time and date when all the members of the Bachelor Project Commission are able to be present. The commission consists of the TU Coach, the Client adviser and one of the Bachelor Project Coordinators. The committee meets to decide the grade right after the final presentation.

Students also often invite friends and family to the final presentation. In general, however, the presence of a larger audience should not (and does not) affect the final presentation any relevant respect. Some teams, however, find that they can present more professionally and convincingly if they are presenting in front of a larger audience.

**Evaluation**: The bachelor's project is evaluated with respect to six dimensions.

- Degree of difficulty
- Research report
- Process (teamwork and application of professional skills)
- Final Product
- Final Report
- Final Presentation

For more information on how the Bachelor Project is evaluated "Appendix 3: Evaluation Scheme for Computer Science Bachelor Project" below.

**Questions?** For additional information, contact the Computer Science Bachelor Project coordinators: Martha Larson (mailto:m.a.larson@tudelft.nl) Felienne Hermans (f.f.j.hermans@tudelft.nl)

## Appendix 1: Learning Objectives of the Computer Science Bachelor Project

Learning objectives: At the end of the Computer Science bachelor project students will have/demonstrate tangible proof that they…

LO1: …can carry out an entire software development cycle with success, from researching solutions through testing the product, in a team of developers addressing a real-world problem.

LO2: …can in effectively, in collaboration with a coach and a client, choose a development strategy and execute a development process according to that strategy.

LO3: …can establish the necessary quality requirements for a product and to carrying out the tests necessary to determine that the product fulfills those requirements.

LO4: …can present a complete and convincing explanation of the development process and the product results.

### Appendix 2: Guidelines for the Final Report

Ten requirements for the formal form of the final report. The report…

1. Must be well structured. Each section, sub-section should have a clear purpose. Paragraphs should open with a topic sentence and the rest of the paragraph should support the topic.

2. Must contain at least 6 sections. Sections should at least include: a foreword, a table of contents, a summary, an introduction, problem definition and problem analysis, the design, the implementations, the conclusions, discuss and recommendations, a conclusion, a reference list. Note that different structures will be more appropriate for different projects.

3. Must include references to the references or other sources that were analyzed during the research phase. Please check that your references are properly and consistently formatted.

4. Must be spell- and grammar-checked. Check that abbreviations are defined at their first use. Check that tables and figures are labeled with captions.

5. Must include a title page. Make sure not to forget: name of the members of the team and the commission (Coach, Client, Bachelor Project Coordinator). Include your student number in the email that you send to your commission, but not in the report.

6. Must include the research report (10 pages) either as an appendix or integrated into the report.

7. Should include the original project description as an appendix.

8. Should be 30-50 pages (plus appendices, if necessary).

9. Must include the first feedback of the SIG, how the team addressed the feedback, and the second feedback of the SIG.

10. The report is a public document. Proprietary information should be kept to the minimum necessary. It should be included in an appendix that can be removed before publishing the report to http://repository.tudelft.nl

Notice that these requirements refer to the formal form of the report, and not to the content of the report. Since each bachelor project team is tackling a unique software development challenge, the content of the report is expected to vary widely from one team to another. Twenty helpful questions that the team should ask themselves about the report in order to understand whether they have written a good report. To understand how these questions differentiate good from not-so-good reports, please refer to examples on: http://repository.tudelft.nl

1. Has the team remembered to include a clear description of the problem that is addressed by the project? Does the description include mention of the immediate context of the project, including the needs and the motivation of the Client?

2. Is the larger context and vision of the project described? What is the potential impact on day-to-day business or business, on people's daily lives, on society at large?

3. Is the solution proposed by the team clearly described?

4. Is the software development methodology chosen by the team clear and well justified?

5. Is the process by which the team arrived at its solution to solution clearly described? Does it reflect the problem analysis that was carried out and the research questions that were addressed? Does the report include the appropriate references?

6. Are the requirements clear? Do they have an appropriate level of detail? Are they prioritized?

7. Is it clear that the team developed success criteria, i.e., did they have a clear picture at the beginning of the project how they would recognize success when they achieved it?

8. Are the system specifications complete? Do they reflect both the big picture and the details needed for implementation? Are UML diagrams used to support understanding of the application? Are they used appropriately (communicate what is essential, but also avoid redundancy)?

9. Does the report include plans for quality control and testing?

10. Does the report allow the reader to understand how the software development methodology was applied? Does the report allow the reader to understand the division of labor among members of the team and how the team communicated (both among themselves and with the Coach and the Client)?

11. Did the team remember to include a complete description of the final product? (Are graphics used appropriately to this end?)

12. Does the report close the loop (i.e., provide evidence that the initial problem was solved)?

13. Does the report include the necessary background information? This should be the information needed by someone with a technical background, but not necessarily one lying directly in the field of the project to understand it? (Or think about the more specific question, "Would the Dean be able to follow it?" http://www.ewi.tudelft.nl/en/the-faculty/dean/)

14. Is something that the team spent a lot of time on not mentioned in the report? Did the team make any important decisions that do not appear in the report?

15. Are there important insights about what the team feels that they should have done differently in retrospect that are not included in the report?

16. Does the team have recommendations for the Client that were not included in the report?

17. Would this report allow other similar teams to understand what was done, in order to extend this project or to create a similar one?

18. Does the report as a whole convey to the reader that the team is has the software development experience necessary in order to carry out a follow-up contract? i.e., that if the team were assigned another software development project by a Client, the team could carry it out as successfully as they carried out this one.

19. Although all bachelor projects are software development projects and follow a particular pattern, in the end every project has a unique aspect to it. Does the report as a whole communicate how this project was special or unique? Are there innovative aspects that should be more clearly emphasized.

20. The report is the "official" record of the bachelor project and is published to http://repository.tudelft.nl. Is the team happy that the report adequately represents the effort that they invested into the project and the solution and insights that they achieved?

**Appendix 3: Evaluation Scheme for Computer Science Bachelor Project**

The TU Delft Computer Science Bachelor Project is evaluated by the Bachelor Project Commission. For a given bachelor project, the Commission consists of three members: the Client (*opdrachtgever*), the TU Coach (*EWI begeleider*), and one of the Computer Science Bachelor Project Coordinators. The Commission meets directly after the final bachelor project presentation, to evaluate the project and decide on the final grade. The commission meeting lasts between 10-20 minutes, although it may last up to 30 minutes in cases where more extensive discussion is necessary. Traditionally, the bachelor project team waits while the commission meets. When the commission meeting has ended, the commission provides feedback to the team and announces the final grade.

The commission decides on the final grade of the bachelor project by considering six evaluation dimensions: Degree of difficulty, Research, Process, Final Product, Final Report and Final Presentation.

During the commission meeting, the Bachelor Project Coordinator requests that members of the commission make statements describing the performance of the bachelor project team along each of these evaluation dimensions. The Bachelor Project Coordinator notes down statements that are accepted by all members of the commission. These statements provide very valuable feedback to the students about their project, and help them to make decisions about how to handle future software development projects. If a member of the commission makes a statement that is not accepted by the other members, that statement is discussed, and reformulated, until it has reached a form that all members accept.

The Bachelor Project is a "real-world project", and each project is unique. For this reason, the statements made by the commission vary from project to project. In general, it is not possible, or desirable, to characterize the success of a bachelor project by choosing from a pre-determined, closed list of possible statements. However, it is expected that the statements made by the Commission address the full scope of the bachelor project. In other words, they must answer at least a set of basic questions. During the commission meeting, the Bachelor Project Coordinator ensures that the set of basic questions have been covered. The basic questions are listed below.

The final step of the commission meeting is to review all statements of the Commission to arrive at a consensus on each of the six dimensions. In order to pass the bachelor project, the project team must achieve adequate performance in all six dimensions. Adequate performance for an evaluation dimension has one of three levels, "-" (falls short of expectations), "∼" (meets expectations), and "+" (exceeds expectations). The final grade for the Bachelor Project is determined by a combination of the levels of performance in all the dimensions. In order to achieve a high grade, all six dimensions must exceed expectations. The highest grades are awarded to teams who end with a product that is ready to be adopted (it is a finished software product, and/or has resulted in a scientific publication).

- The TU Coach should keep a careful eye out for problems during the course of the project, and take action if a risk of failing one of the dimensions is observed (including contacting the Bachelor Project Coordinators). The Client and the team themselves also share responsibility for taking early action if the project appears to be at risk. Possible problems include: one or more members of the team not pulling their weight, communication issues between the team members or with the Client.
- Note that by convention, the highest achievable grade in the Computer Science Bachelor Project is 9. The TU Coach should keep a careful eye out for outstanding teams who would require this

convention to be overthrown. The Bachelor Project Coordinators should be notified so that the past precedents can be compared systematically with the current performance of the team.
- More than one person from the side of the Client can take part in the commission meeting, in order to provide a complete perspective on the experience of the project team. However, the input of the Client receives the same weight, independently of how many Client representatives attend the commission meeting.

In case the Commission would not converge on consensus, then a second meeting is called that includes both Bachelor Coordinators. In practice, however, enough communication has taken place during the course of the project between the TU Coach, Client, and, if necessary, the Bachelor Project Coordinator, than no large disparity arises in the conclusions of the members of the Commission during the commission meeting.

**Basic Questions used to Evaluation the TU Delft Computer Science Bachelor Project**

**Difficulty**
- Do existing (e.g., off-the-shelf) solutions to the problem already exist?
- Does the project have multiple facets? Were many decisions required?
- Did the project require that the team make use of a diverse skill set?

**Research**
- Were the references appropriate and complete?
- Did the research cover the possible alternatives for major decisions about approach, design, and implementation?
- Were the project decisions grounded in a comparative analysis of the alternative solutions?

**Process**
- Did the process run smoothly along schedule? Did the communication between team members support quick resolution of unexpected challenges?
- Did the team make good use of software development methodology?
- Did the team work independently? Was the team pro-active in seeking help when needed?

**Product**
- Did the system fulfill the original system specifications? Was the product well tested?
- Was the code of high quality? Did the team take advantage of the SIG input?
- Will/can the Client use the product (has the Client's goal been achieved)? Does it advance the state of the art or bring a new application to market?

**Report**
- Was the report well structured and clearly written?
- Did the report present a complete picture of the project (i.e., no important phases in the project trajectory where omitted)?
- Did the report provide motivation that the project was successful?

**Presentation**
- Was the presentation well structured and clear (including demo)?
- Did the presentation convince the audience that the project was substantial, well motivated, and successful?
- Did the team demonstrate their mastery of the material during the Q&A.