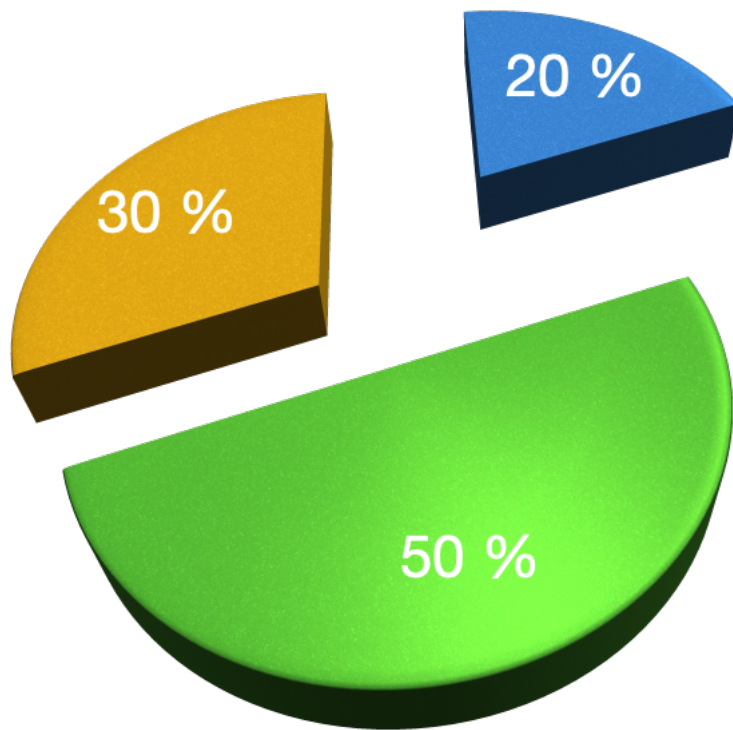


Mikrocomputertechnik 1

Thomas Stingl (Airbus)

Projektarbeit

Projektarbeit - Übersicht



Pitch



Programmenwurf



Präsentation

Aufgabenstellung

Entwickeln Sie gemeinsam in Gruppen Ihr Mikrocontroller-Projekt. Dabei stehen Ihnen alle Mittel zur Verfügung. Dabei sind zwingend folgende Meilensteine und Reviews in Ihre Projektplanung einzuarbeiten:

- Startup - Pitch (zu Beginn festes Datum für alle); wöchentliche Berichte (Wöchentlich 2 Tage vor Vorlesung)
- Abgabe der Projektarbeit inklusive Dokumentation, Code, Engineering Folder, etc.
- Vorstellung und Präsentation Ihrer Arbeit

Zeigen Sie im Rahmen des Ihnen gewählten Entwicklungsprozesses V-Modell/Agile, wie sie Ihr Projekt durch die verschiedenen Entwicklungsphasen von Requirements Engineering, Design Descriptions, Test Spezifikation und Reports bringen. Zeigen Sie auch auf, wie Sie diese Ziele erreichen wollen, aus welchen Teilen sich Ihr Projekt zusammen setzt, welche Standards oder Regeln sie anwenden wollen. Vergessen Sie neben Ihren MCT technischen Focus nicht die Kosten, Risiken, Opportunities und den Schedule - in einem Wort zusammengefasst - Management.

Projektarbeit - Pitch

Aufgabenstellung:

Wählen sie aus folgenden zwei Szenarios wie sie Ihr Projekt darstellen (oder Moderen „pitchen“) wollen:

- Investorensuche
- Überzeuge den Chef

Versuchen Sie dabei ihr Projekt als Ganzes zu sehen und wenden sie Erkenntnisse aus anderen Bereichen Ihres Studiums an (z.B. Projektplanungen, Informatik, etc.) an.

Gehen Sie dabei auf jeden Fall auch Mikrocontroller relevante Probleme ein und überzeugen Sie durch Technik. Zeigen Sie wie sie Ihr Projekt unter den gegenbenden Rahmenbedingung (Zeit, Kosten, ...) lösen wollen. D.h. teilen Sie Ihr Projekt in Phasen ein, definieren sie final Dates und Cut-Offs für Entscheidungen und nicht zuletzt zeigen Sie Ihr Vorgehen aus der Vogelperspektive auf. Der Vortragsstil ist frei wählbar ebenso die eingesetzten Mittel.

Hinweis: Beachten Sie in jedem Fall die Kosten - diese sollten von Ihnen selbst tragbar sein!

Lösungsskizze:

Aus Mikrocontroller Sicht sind die in der Vorlesung Besprochenen Themen abzuklären. Dabei muss das Ziel mit der Gewählten Mikrocontroller Architektur möglich sein. Verschiede I/O die für die Umsetzung gewählt wurde muss im zeitlichen Rahmen und in der Teamstärke umsetzbar sein.

Im Rahmen der Präsentation einer Investorensuche soll eine kleine Marktanalyse bzw. eine Bewertung etwaiger Konkurrenzprodukte erfolgen. Dabei soll jedoch immer die Technik in Vordergrund stehen.

Im Rahmen der Präsentation um den Chef zu überzeugen, sollte auf die Gründe und Möglichkeiten eingegangen werden die ein Unternehmen durch den Einsatz Ihrer Arbeit hat. D.h. Kostenreduktion, Vermeidung unnötiger Aufwendungen.

Ziel bei dieser Übung ist, dass sich das Team findet und sich kennen lernt. Wichtig ist auch, dass die Motivation durch das selbst gewählte Projekt erkennbar ist aber auch die Vernunft über das was erreicht werden kann abgeschätzt wird. Selbst wenn am ende nicht der volle Funktionsumfang erreicht wird, muss das nicht negativ bewerte werden, denn wir trainieren unsere Selbsteinschätzung jeden Tag aufs Neue.

Bewertungspunkte:

- Wurde das Projekt klar beschrieben
- Wurde der Entwicklungsprozess grob dargestellt
- Wurden technischen Probleme erkannt
- Gibt es einen Zeitplan?
- Risk&Opportunites erkannt?
- Canvas verstanden und erklärt?

Projektarbeit - Wöchentliche Berichte

Aufgabenstellung:

Dokumentieren Sie die Fortschritte Ihres Projektes.

- Zeigen Sie auf, ob der Schedule eingehalten werden kann.
- Zeigen Sie die Entwicklung Ihrer Risiken und Opportunitäten. Konzentrieren Sie sich auf Schlüssentscheidungen im Design.
- Zeigen Sie die Probleme und Schwierigkeiten im Projekt. Bedenken Sie, dass viele Projekte scheitern, weil selbst erfahrene System-Ingenieure nicht offen über die Probleme berichten wollen, aus Angst vor Management-unnötigem Reporting zu bekommen.
- Zeigen Sie offen und ehrlich, was Sie im Team brauchen, um Schwierigkeiten zu überwinden - machen Sie also Vorschläge, bevor Ihnen die Vorschläge abgenommen werden.

Achtung - das Schreiben und Fertigstellen eines Berichtes sollte nicht länger als 10 Minuten in Anspruch nehmen. Wählen Sie da

Lösungsskizze:

Der wöchentliche Bericht ist ein wichtiges Medium, das die Studenten während Ihrer Projektarbeit unterstützen soll, den Kursleiter auf Probleme aufmerksam zu machen. Falls Probleme entstehen, kann die Vorlesung diese Themen aufgreifen und abarbeiten.

Bewertungspunkte:

- Wurde das Erreichte klar und kurz formuliert
- Wurde auf die notwendigen Themen erkannt und berichtet
- Wurde die Risiken bzw. Opportunitäten neu bewertet
- Wurde bei technischen Problemen Alternativen erkannt und bewertet
- Wurde der Schedule im Verlauf kontrolliert und ggf. angepasst

Projektarbeit - Pitch Presentation

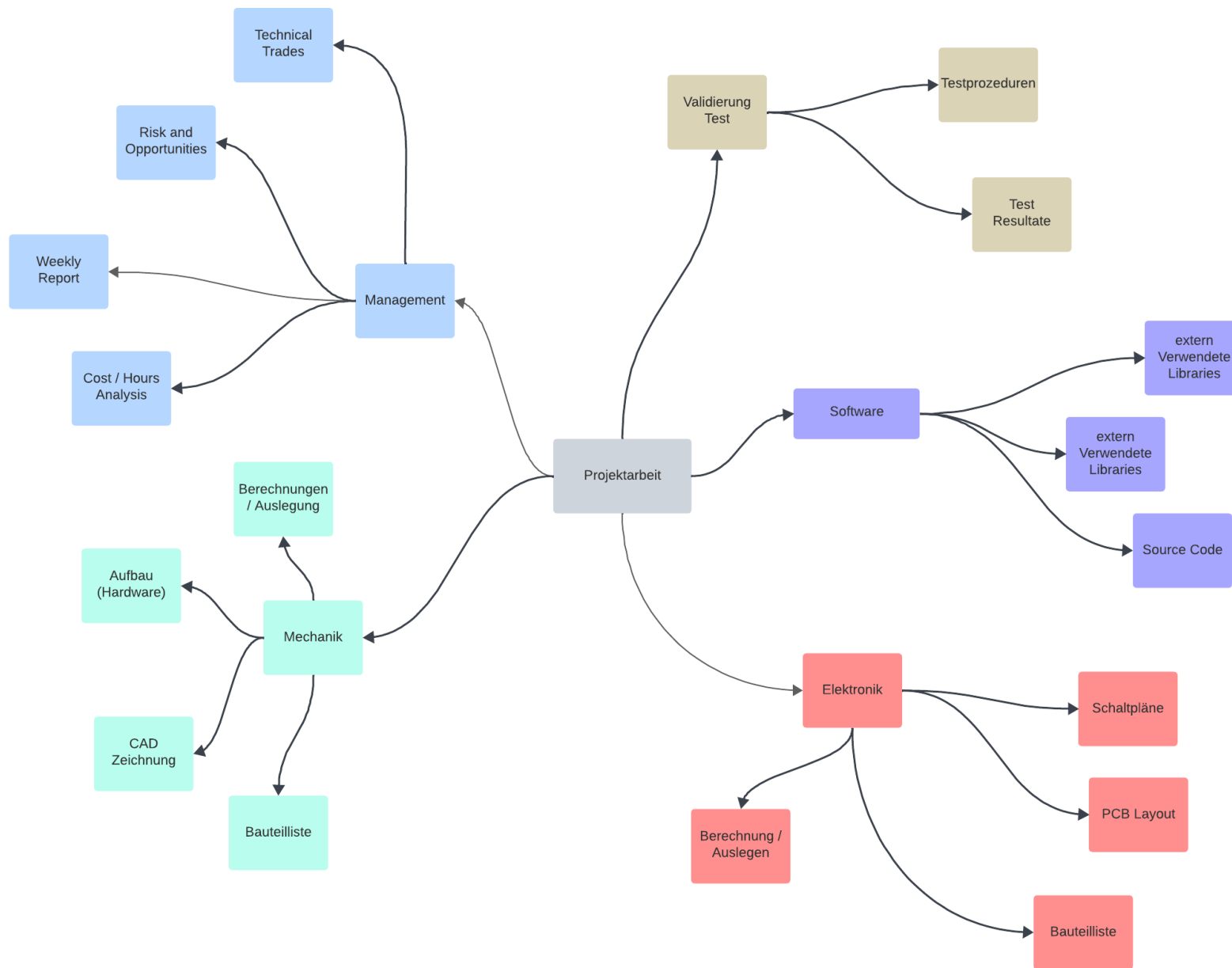
Duration: max. 6 Minutes
(4 minutes for Questions)

Define the situation you want to present - either
Marketing / Investor Pitch
Convince my Boss / Scenario

Consider what your counterpart is interested in?

Must Have - Project Canvas!

Tools: Presentation, Slides, Whatever you need and think is useful!!!



Pitch

Who is the audience and what are their interest?

What your main message with your product?

Why YOU!

Trade costs, seriousness and marketing e.g. Risks

Engineering Phase

Write Requirements (SMART) consider Testmethodes

Start implementing your Document Tree

Trade different solutions different possibilities and techniques.

Finalize setup for development tools

Collaborate together as much as possible

Monitor Risks and Opportunities

Design Phase

Start your development at architectural level by diagrams in-line with your Requirements

Consider that everything need to be tested

No Problem if you want to change or modify requirements that are not suitable

Continue collaboration and keep an eye on schedule, costs and ...

Finalize this phase with specifying SW/Electrical requirements

Implementation Phase

Consider help from your project partners; double check code, etc.

Apply your coding rules and standards and check (Quality!)

Collaboration can be an added value but not strictly required anymore

Write your test procedures

Start coding requirements and do integration tests

Validation Phase

Document test effort

Consider Error Management - e.g. Document errors and use them as an update reason

Be strict on Configuraiton Control on what you test how

Check your requirement coverage

Final Data Pack Presentation

Think that a failure can be a win

Prepare your Lessons Learned

Prepare your final presentation of your projet

Projektarbeit - Programmentwurf

Aufgabenstellung:

Ihre gesamte Entwicklung darf in eine schriftliche Form gebracht werden. In Firmen wird das oftmals mit einem Review bezeichnet (PDR - Preliminary Design Review, CDR - Critical Design Review, etc.) Da bei unserer MCT Projekt ein fast kompletter Lifecycle darzustellen ist stellen Sie im Rahmen des Programmentwurfs ein kombiniertes PDR und (fast) CDR dar.

Die Dokumente dürfen auch in online repositories zur Verfügung gestellt werden. Bitte beachten Sie, dass diese evtl. mithilfe von Doxygen o.ä. extrahierbar sind.

Vorgegeben ist folgender Rahmen (nächstes Slide) in dem sie Ihre Arbeit abgeben sollen:

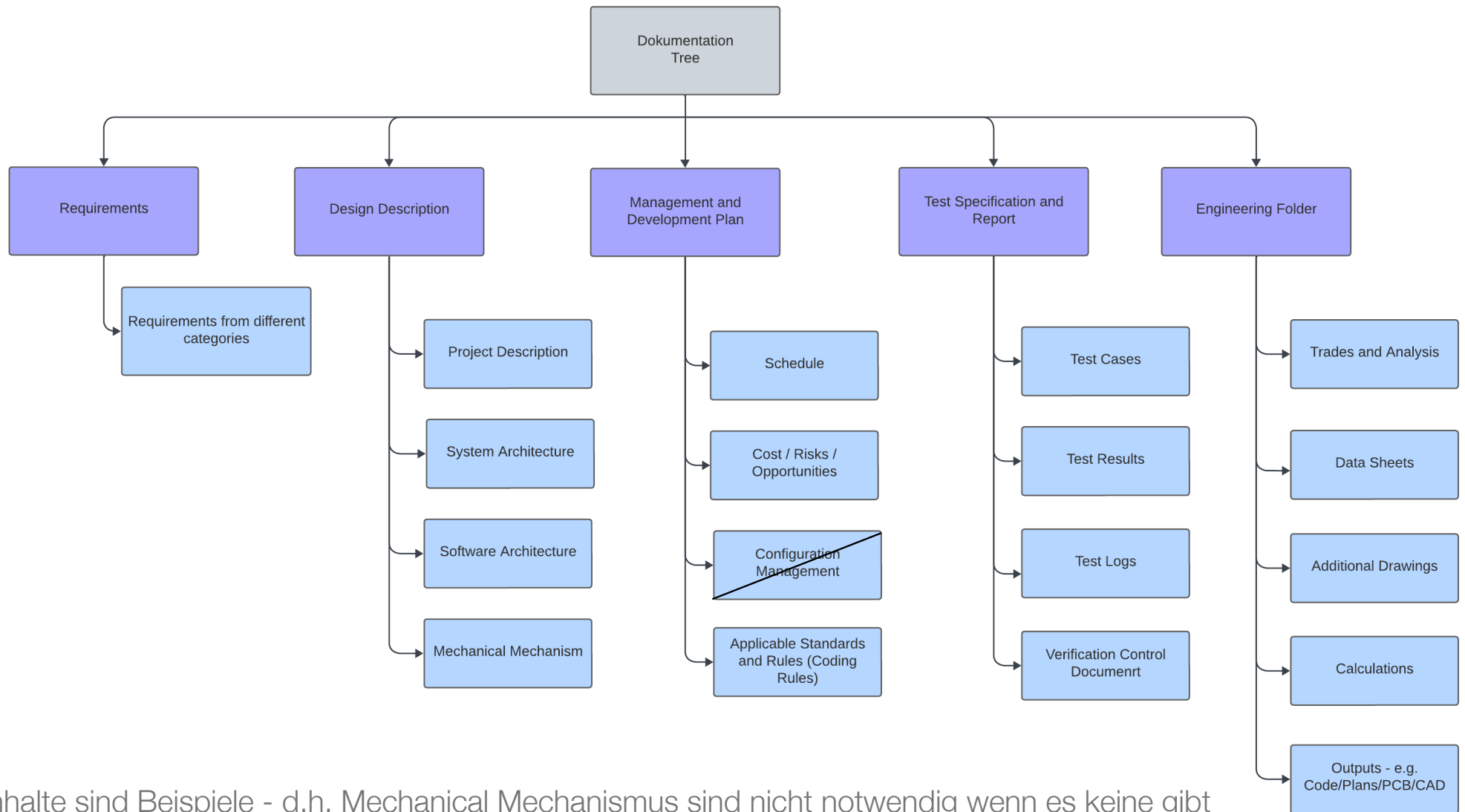
Achtung - alles muss als PDFs (alt. XLS, WORD, png, jpg...) in einem Archiv (ZIP) abgeben werden.

Lösungsskizze:

Der Rahmen der Aufgabenstellung

Bewertungspunkte:

- Lessons Learned - wurden im Rahmen der Entwicklung klar gezeigt was gut lief bzw. was weniger gut lief
- Vollständig d.h. zeigt die Dokumentation alle teile des Entwicklungszuyklus aus
- Treffende Projektplanung
- Nachvollziehbare Dokumentation (Roter Faden)
- Überzeugende System. / Sub-Systemarchitektur
- MCT optimiertes (Code) Design
- Abgeschlossenes V-Modell
- Formalia (Zitierweise, Korrekte Anwendung von Grammatik, gute Lesbarkeit)



Inhalte sind Beispiele - d.h. Mechanical Mechanismus sind nicht notwendig wenn es keine gibt

Projektarbeit - Präsentation

Aufgabenstellung:

Stellen Sie Ihre Arbeit vor!

Zeigen Sie offen in der Gruppe und gegenüber den anderen Gruppen was gut lief und was nicht.

Gemeinsam (viele Gruppen) können Sie so von den Anderen Arbeiten, deren Lessons Learned profitieren. Zeigen Sie was Ihrer Meinung nach besonders war in Ihrem Ablauf. Das können z.B. ein Testframework, eine Schedulability Analyse, ein Interface zu einem zweiten Controller für WWW Anbindung, etc. sein.

In einer Firma ist es besonders wichtig, dass Sie wissen was Sie tun und erreichen können und auch was ein Anderer schon gemacht hat - Networking. Diese offene Kommunikation in Form einer Präsentation kann der ganzen Abteilung/Firma (dem Kurs) helfen effizienter zu werden indem Sie Ihre Ergebnisse / Erkenntnisse teilen.

Ein Zurückhalten von Fehlschlägen sowie eine nicht offene Kommunikation darüber kann die Effizienz einer Firma deutlich verschlechtern. Zeigen Sie hier dass Sie offen dafür sind mit Ihren Kollegen technische Lösungen und Probleme offen anzusprechen und auch teilen können.

Lösungsskizze:

Lessons Learned stehen im Vordergrund und wie damit im Verlauf des Projektes umgegangen wurde. Hat das Team das erkannt was bei Ihnen besonders lief bzw. zum Nutzen der anderen sein kann und das entsprechend aufbereitet.

Bewertungspunkte:

- Lessons Learned - stehen klar im Vordergrund
- Was hat gut funktioniert?
- Was hat weniger gut funktioniert? - und die Analyse dessen
- Wurde das Special Topic erkannt und aufbereitet

Projektarbeit - Final Presentation

Duration: max. 15
(5 minutes for Questions)

Present your key technical trades and decisions!
Present the different difficulties you were confronted with!
Present things that went wrong and why!
Present things you can do better next time!

Consider what your counterpart is interested in?

Tools: Presentation, Slides, Whatever you need and think is useful!!!

Dokumentation Beispiel

Code Documentation - General Info

```
/******  
* Copyright (C) 2012 by Matteo Franchin *  
* * * * *  
* This file is part of Box. *  
* * * * *  
* Box is free software: you can redistribute it and/or modify it *  
* under the terms of the GNU Lesser General Public License as published *  
* by the Free Software Foundation, either version 3 of the License, or *  
* (at your option) any later version. *  
* * * * *  
* Box is distributed in the hope that it will be useful, *  
* but WITHOUT ANY WARRANTY; without even the implied warranty of *  
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the *  
* GNU Lesser General Public License for more details. *  
* * * * *  
* You should have received a copy of the GNU Lesser General Public *  
* License along with Box. If not, see <http://www.gnu.org/licenses/>. *  
*****/
```

Code Documentation - File

```
/**
 * @file doxygen_c.h
 * @author My Self
 * @date 9 Sep 2012
 * @brief File containing example of doxygen usage for quick reference.
 *
 * Here typically goes a more extensive explanation of what the header
 * defines. Doxygens tags are words preceeded by either a backslash @\
 * or by an at symbol @@.
 * @see http://www.stack.nl/~dimitri/doxygen/docblocks.html
 * @see http://www.stack.nl/~dimitri/doxygen/commands.html
 */
```

Code Documentation - Function

```
/**
 * @brief Example showing how to document a function with Doxygen.
 *
 * Description of what the function does. This part may refer to the parameters
 * of the function, like @p param1 or @p param2. A word of code can also be
 * inserted like @c this which is equivalent to <tt>this</tt> and can be useful
 * to say that the function returns a @c void or an @c int. If you want to have
 * more than one word in typewriter font, then just use @<tt>.
 * We can also include text verbatim,
 * @verbatim like this@endverbatim
 * Sometimes it is also convenient to include an example of usage:
 * @code
 * BoxStruct *out = Box_The_Function_Name(param1, param2);
 * printf("something...\n");
 * @endcode
 * Or,
 * @code{.py}
 * pyval = python_func(arg1, arg2)
 * print pyval
 * @endcode
 * when the language is not the one used in the current source file (but
 * <b>be careful</b> as this may be supported only by recent versions
 * of Doxygen). By the way, <b>this is how you write bold text</b> or,
 * if it is just one word, then you can just do @b this.
 * @param param1 Description of the first parameter of the function.
 * @param param2 The second one, which follows @p param1.
 * @return Describe what the function returns.
 * @see Box_The_Second_Function
 * @see Box_The_Last_One
 * @see http://website/
 * @note Something to note.
 * @warning Warning.
 */
```

Code Documentation - Types

```
/**
 * @brief Use brief, otherwise the index won't have a brief explanation.
 *
 * Detailed explanation.
 */
typedef enum BoxEnum_enum {
    BOXENUM_FIRST, /**< Some documentation for first. */
    BOXENUM_SECOND, /**< Some documentation for second. */
    BOXENUM_ETC    /**< Etc. */
} BoxEnum;

/**
 * @brief Use brief, otherwise the index won't have a brief explanation.
 *
 * Detailed explanation.
 */
typedef struct BoxStruct_struct {
    int a;    /**< Some documentation for the member BoxStruct#a. */
    int b;    /**< Some documentation for the member BoxStruct#b. */
    double c; /**< Etc. */
} BoxStruct;
```