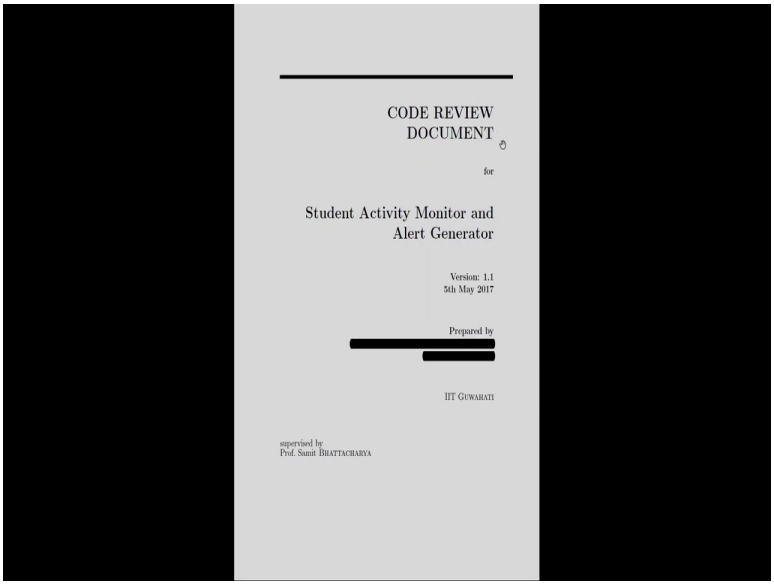
Design and Implementation of Human – Computer Interfaces
Prof. Dr. Samit Bhattacharya
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati

Lecture: 30
Code Review Case Study

Hello and welcome to the NPTEL MOOCS course on design and implementation of human computer interfaces. We will continue our discussion on review based code testing as lecture number 26 in this lecture in this continuation of lecture number 26 we will go through one case study for review based code testing. Just to recollect earlier we discussed about different code testing methods.

We talked about review based code testing where we have seen two broad approaches one is walkthrough another one is inspection. In this case study we will primarily focus on inspection based code testing. Let us see the case study.

(Refer Slide Time: 01:32)

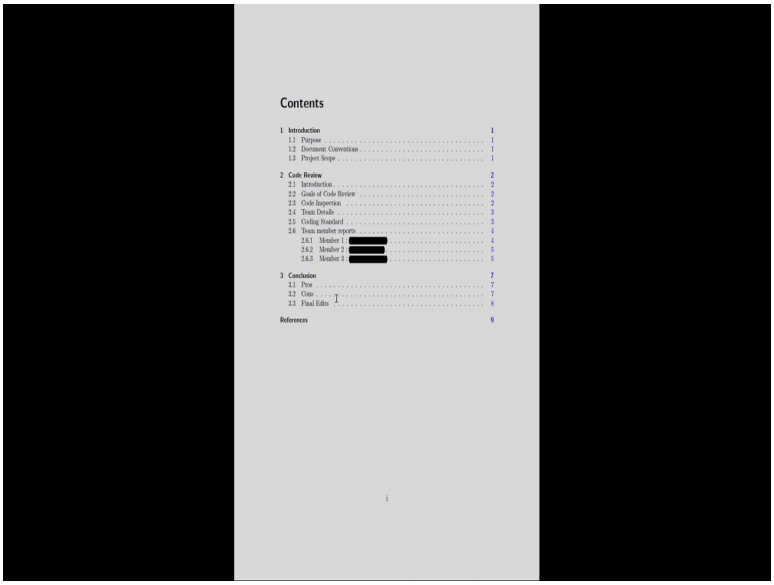


The case is development of a system which is meant for monitoring of Student Activity in a classroom and also the system is supposed to generate some alert if there is some anomaly found in the monitoring of Student Activities. So, this is the cover page remember that after this code review phase we are expected to develop a review document. So, in this case study we will go

through one such document this is the cover page of the document code review document for name of the system as shown here.

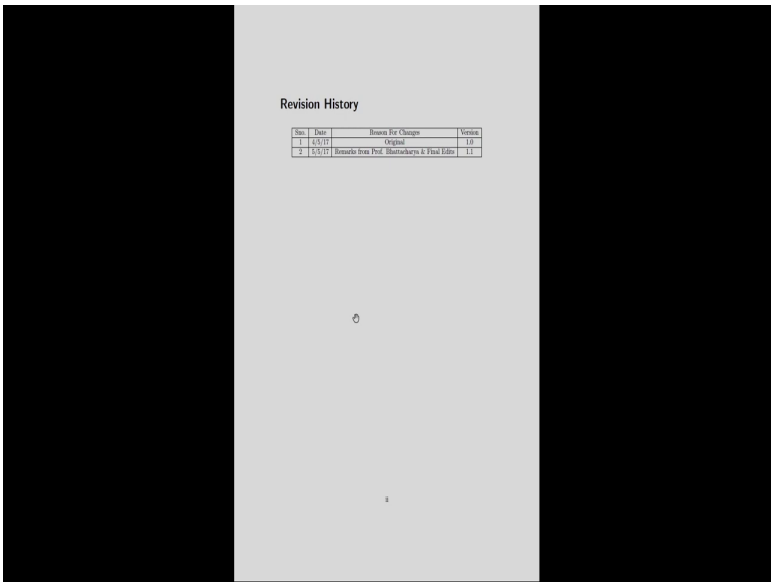
As we have seen earlier the version of the document ideally should be mentioned along with the date of creation prepared by the name of the persons who prepared the document. Additional information are provided here that the place at equity supervised by the person's name these are optional of course but the cover page should contain the name of the system version date and name of the persons who created the document. These are essential things that should be part of the cover page.

(Refer Slide Time: 03:05)



Next comes a table of contents which includes the sections as well as the sub sections that are part of the sections like here in this document there are three main sections introduction code review and conclusion each having some subsections as listed here.

(Refer Slide Time: 03:24)



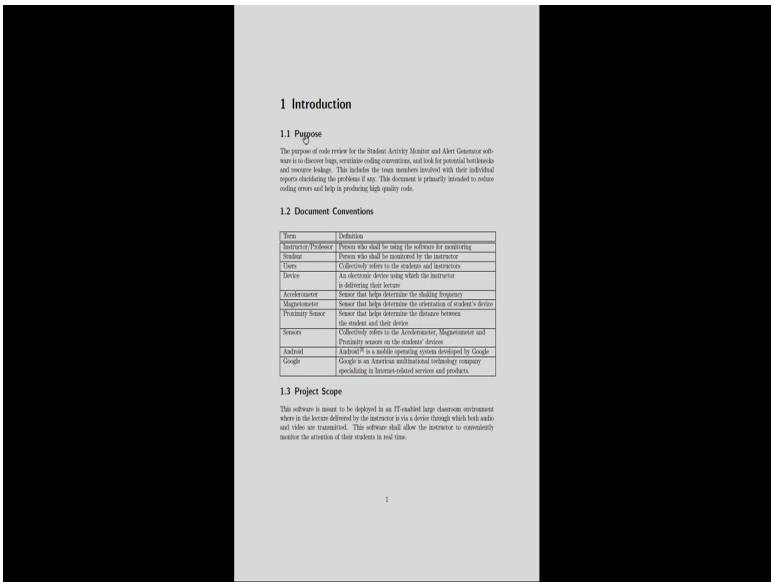
Revision History

Sno.	Date	Reason For Changes	Version
1	4/5/17	Original	1.0
2	5/5/17	Remarks from Prof. Bhattacharya & Final Edits	1.1

ii

Optionally it is advisable to include some revision history like shown here that the first version was created on so and so date named version 1.0 then it was reviewed and refined based on the comments received from the supervisor. And revision had been created so and so date and the revision version number is given as 1.1 which is the current version. So, this revision history ideally kept as part of the document.

(Refer Slide Time: 04:02)



1 Introduction

1.1 Purpose

The purpose of code review for the Student Activity Monitor and Alert Generator software is to discover bugs, scrutinize coding conventions, and look for potential bottlenecks and resource leakage. This includes the team members involved with their individual reports elucidating the problems if any. This document is primarily intended to reduce coding errors and help in producing high quality code.

1.2 Document Conventions

Term	Definition
Instructor/Professor	Person who shall be using the software for monitoring
Student	Person who shall be monitored by the instructor
Users	Collectively refers to the students and instructors
Device	An electronic device using which the instructor is delivering their lecture
Accelerometer	Sensor that helps determine the shaking frequency
Magnetometer	Sensor that helps determine the orientation of student's device
Proximity Sensor	Sensor that helps determine the distance between the student and their device
Sensors	Collectively refers to the Accelerometer, Magnetometer and Proximity sensors on the students' devices
Android	Android™ is a mobile operating system developed by Google
Google	Google is an American multinational technology company specializing in Internet-related services and products.

1.3 Project Scope

This software is meant to be deployed in an IT-enabled large classroom environment where in the lecture delivered by the instructor is via a device through which both audio and video are transmitted. This software shall allow the instructor to conveniently monitor the attention of their students in real time.

1

Then the introduction part where it is preferable that the purpose of this document, conventions used in the document scope of the project all these things are mentioned. In the purpose what can be written here what is written is the purpose of code review for the student activity monitor and

alert generation software is to discover bugs, scrutinize coding conventions and look for potential bottlenecks and resource leakage.

This includes the team members involved with their individual reports elucidating the problems if any. This document is primarily intended to reduce coding errors and help in producing high quality code. So, that is the purpose of code review document basically to reduce errors. Then the subsequent part of the document made use of some terms. So, those are listed in this section document conventions like whenever the term instructor or Professor is used it is defined as a person who shall be using the software for monitoring.

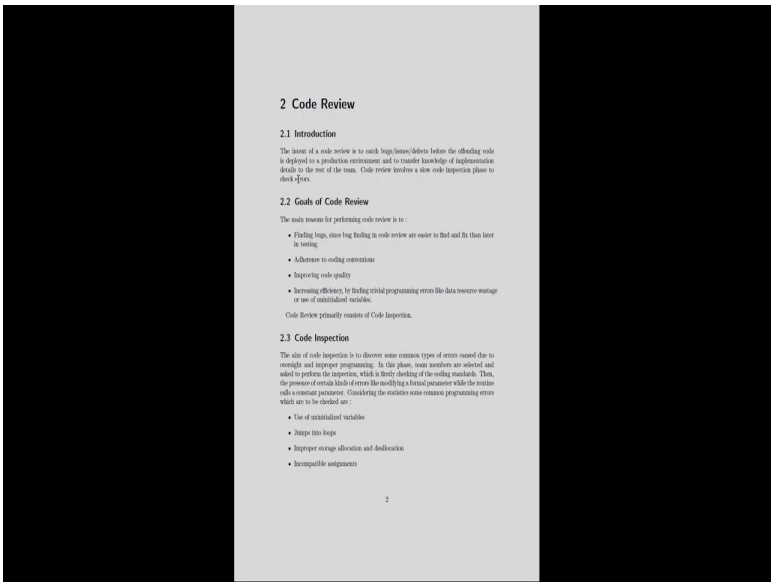
Then student term indicates person who shall be monitored by the instructor. Users are collectively refers to the students and instructors both categories device an electronic device using which the instructor is delivering their lecture. Accelerometer sensor that helps determine the second frequency of a mobile device. Magnetometer sensor that helps determine the orientation of students mobile device, proximity sensor, sensor that helps determine the distance between the student and their device.

The term sensors is defined as this term differs collectively to the accelerometer magnetometer and proximity sensors on the student's mobile device. Android is the mobile operating system developed by Google. Google is an American multinational technology company. So, all the details are mentioned here. It may be helpful to know that this is a part of a larger system. So, in the larger system it is assumed that there is an ICT enabled classroom where students carry their own mobile devices and instructor also carry a mobile device.

And these device sensors are being mentioned in this document. The whole system is implemented using mobile devices. The next subsection talks about the scope of the project that is the scope of the software the software is meant to be deployed in an IT enabled large classroom environment where in the lecture delivered by the instructor is via a device through which both audio and video are transmitted.

The software shall allow the instructor to conveniently monitor the attention of their students in real time in brief the scope is mentioned here. These are the subsections of the introduction section next comes the core component of this document that is the code review part.

(Refer Slide Time: 07:18)



2 Code Review

2.1 Introduction

The intent of a code review is to catch bugs/issues/defects before the offending code is deployed to a production environment and to transfer knowledge of implementation details to the rest of the team. Code review involves a slow code inspection phase to check errors.

2.2 Goals of Code Review

The main reasons for performing code review is to :

- Finding bugs, since bug finding in code review are easier to find and fix than later in testing
- Adherence to coding conventions
- Improving code quality
- Increasing efficiency, by finding trivial programming errors like data resource wastage or use of uninitialized variables.

Code Review primarily consists of Code Inspection.

2.3 Code Inspection

The aim of code inspection is to discover some common types of errors caused due to oversight and improper programming. In this phase, team members are selected and asked to perform the inspection, which is firstly checking of the coding standards. Then, the presence of certain kinds of errors like modifying a formal parameter while the routine calls a constant parameter. Considering the statistics some common programming errors which are to be checked are :

- Use of uninitialized variables
- Jumps into loops
- Improper storage allocation and deallocation
- Incompatible assignments

2

It starts with some introduction section the intent of a code review is to catch bugs issues defects before the offending code is deployed to a production environment and to transfer knowledge of implementation details to the rest of the team. Code review involves a slow code inspection phase to check errors. So, here it is clearly mentioned that this particular review document contains a code review procedure based on code inspection method.

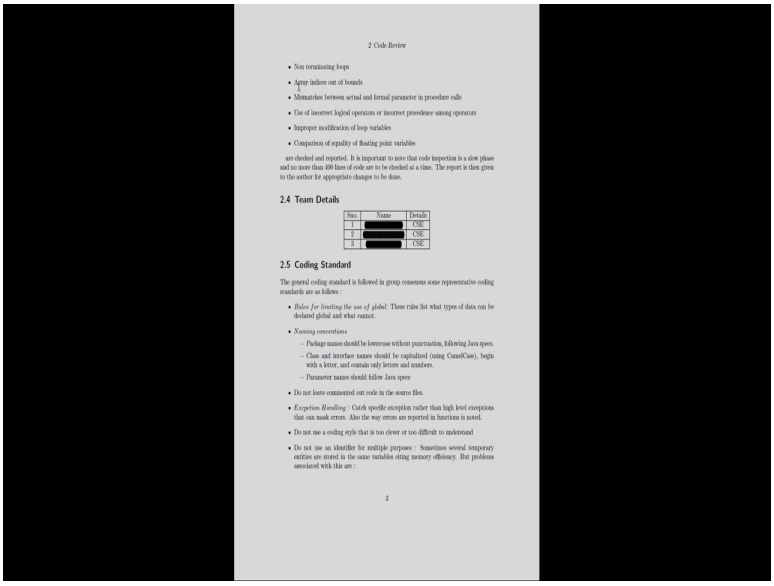
Then the goals of code review the main reasons for performing code review is to finding bugs since bug finding in code review are easier to find and fix than letter in testing phase. Adherence to coding conventions improving code quality increasing efficiency by finding trivial programming errors like data resource wastage or use of uninitialized variables, code review primarily consists of code inspection as per this document. Of course that is not a general statement we can perform review either through a walkthrough method or an inspection method.

Here in this document it is mentioned that the review has been performed using an inspection method. Then it talks about code inspection the aim of code inspection is to discover some common types of Errors caused due to oversight and improper programming. In this phase team

members are selected and asked to perform the inspection which is firstly checking of the coding standards then the presence of certain kinds of Errors like modifying a formal parameter while the routine calls a constant parameter.

Considering the statistics some common programming errors which are to be checked are use of uninitialized variables, jumps into Loops, improper storage allocation and deallocation incompatible assignments.

(Refer Slide Time: 09:32)



2 Code Review

- Non terminating loops
- Array indices out of bounds
- Mismatches between actual and formal parameter in procedure calls
- Use of incorrect logical operators or incorrect precedence among operators
- Improper modification of loop variables
- Comparison of equality of floating point variables

are checked and reported. It is important to note that code inspection is a slow phase and no more than 400 lines of code are to be checked at a time. The report is then given to the author for appropriate changes to be done.

2.4 Team Details

Sno.	Name	Details
1	[redacted]	CSE
2	[redacted]	CSE
3	[redacted]	CSE

2.5 Coding Standard

The general coding standard is followed in group consensus some representative coding standards are as follows :

- *Rules for limiting the use of global*: These rules list what types of data can be declared global and what cannot.
- *Naming conventions*
  - Package names should be lowercase without punctuation, following Java specs.
  - Class and interface names should be capitalized (using CamelCase), begin with a letter, and contain only letters and numbers.
  - Parameter names should follow Java specs
- Do not leave commented out code in the source files.
- *Exception Handling* : Catch specific exception rather than high level exceptions that can mask errors. Also the way errors are reported in functions is noted.
- Do not use a coding style that is too clever or too difficult to understand
- Do not use an identifier for multiple purposes : Sometimes several temporary entities are stored in the same variables citing memory efficiency. But problems associated with this are :

3

Non-terminating loops, array indices out of bound, mismatches between actual and formal parameters in procedure calls, use of incorrect logical operators or incorrect precedence among operators, improper modification of loop variables, comparison of equality of floating Point variables. Recollect during our discussion in the lecture we mentioned that companies keep some statistics of common errors that generally occur while implementing some system.

So, those are essentially used to create a checklist against which a code can be reviewed some of those are mentioned in this list. And these are checked and reported it is important to note that code inspection is a slow phase and no more than 400 lines of code are to be checked at a time the report is then given to the other members or to the developer of this document for appropriate change to be done. So, this report is generated by the team and given to the developer.

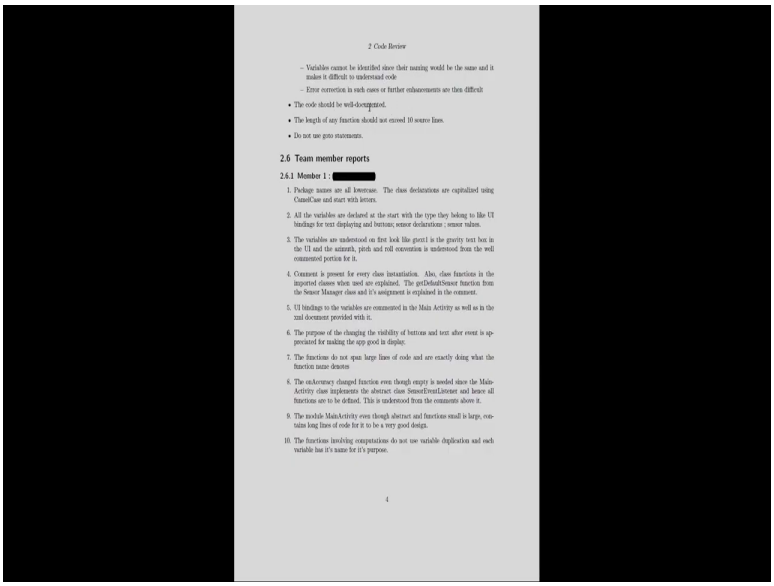
So, that code can be modified. Next the document mentions the team details name and some other section details next comes coding standard. The general coding standard is followed in group consensus some representative coding standards are as follows. Rules for limiting the use of global variables these rules list what type of data can be declared Global and what cannot. Naming conventions package names should be lowercase without punctuations following JavaScript.

So, this is specific to this document class and interface names should be capitalized using camel case begin with a letter and contain only letters and numbers, parameter names should follow javascripts. It can be noted here is that these are some of the standards mentioned in this document but that need not be followed everywhere but some standards should be followed. Do not leave commented out code in the source files.

Exception handling catch specific exception rather than high level exceptions that can mask errors also the way errors are reported in function is noted. Do not use a coding style that is too clever or too difficult to understand. Do not use an identifier for multiple purposes sometimes several temporary entities are stored in the same variable citing memory efficiency but problems associated with this practice are variables cannot be identified since their naming would be the same and it makes it difficult to understand code.

That means we cannot assign them proper names. Error correction in such cases or further enhancements are then difficult. The code becomes difficult to maintain manage and several issues crop up.

(Refer Slide Time: 13:07)



2 Code Review

– Variables cannot be identified since their naming would be the same and it makes it difficult to understand code
– Error correction in such cases or further enhancements are then difficult

- The code should be well-documented.
- The length of any function should not exceed 10 source lines.
- Do not use goto statements.

2.6 Team member reports

2.6.1 Member 1: [redacted]

1. Package names are all lowercase. The class declarations are capitalized using CamelCase and start with letters.
2. All the variables are declared at the start with the type they belong to like UI bindings for text displaying and buttons; sensor declarations ; sensor values.
3. The variables are understood on first look like gtext1 is the gravity text box in the UI and the azimuth, pitch and roll convention is understood from the well commented portion for it.
4. Comment is present for every class instantiation. Also, class functions in the imported classes when used are explained. The getDefaultSensor function from the Sensor Manager class and it's assignment is explained in the comment.
5. UI bindings to the variables are commented in the Main Activity as well as in the xml document provided with it.
6. The purpose of the changing the visibility of buttons and text after event is appreciated for making the app good in display.
7. The functions do not span large lines of code and are exactly doing what the function name denotes
8. The onAccuracy changed function even though empty is needed since the MainActivity class implements the abstract class SensorEventListener and hence all functions are to be defined. This is understood from the comments above it.
9. The module MainActivity even though abstract and functions small is large, contains long lines of code for it to be a very good design.
10. The functions involving computations do not use variable duplication and each variable has it's name for it's purpose.

4

The code should be well documented the length of any function should not exceed 10 Source lines, do not use go to statements. These are some of the conventions against which the code is checked for the purpose of creating the this particular code review document. Some of these we already mentioned in the lecture if you may recollect. Next comes the actual reports obtained from the from the team members.

So, there were three team members accordingly three reports were generated. Member one listed the observations after going through the code and these observations are package names are all lower case the class declarations are capitalized using camel case and start with letters. All the variables are declared at the start with the type they belong to like UI binding for text displaying and buttons censored declarations sensor values.

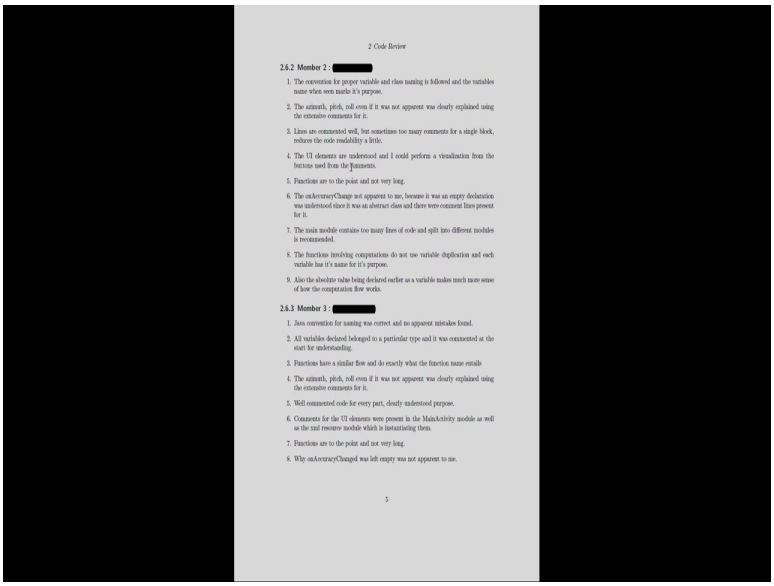
The variables are understood on first Loop like G text 1 is the gravity text box in the UI and the Azimuth pitch and role convention is understood from the well commented person for it. Comment is present for every class instantiation also class functions in the imported classes when used are explained the get default sensor function from the sensor manager class and its assignment is explained in the comment.

UI binding to the variables are commented in the main activity as well as in the XML document provided with it. The purpose of the changing of the visibility of buttons and text after event is

appreciated for making the app good in display of course during inspection there is no actual execution of the code but from the code this particular observation was made by this evaluator the functions do not span large number of lines of code and are exactly doing what the function name denotes.

The on accuracy changed function even though empty is needed since the main activity class implements the abstract class sensor event listener and hence all functions are to be defined this is understood from the comments above it. The module main activity even though abstract and functions are small but the main activity is large contains long lines of codes for it to be a very good design. The functions involving computations do not use variables variable duplications and each variable has its name for its purpose. In other words each variable name reveals its purpose.

(Refer Slide Time: 16:12)



2 Code Review

2.6.2 Member 2 : [redacted]

1. The convention for proper variable and class naming is followed and the variables name when seen marks it's purpose.
2. The azimuth, pitch, roll even if it was not apparent was clearly explained using the extensive comments for it.
3. Lines are commented well, but sometimes too many comments for a single block, reduces the code readability a little.
4. The UI elements are understood and I could perform a visualisation from the buttons used from the comments.
5. Functions are to the point and not very long.
6. The onAccuracyChange not apparent to me, because it was an empty declaration was understood since it was an abstract class and there were comment lines present for it.
7. The main module contains too many lines of code and split into different modules is recommended.
8. The functions involving computations do not use variable duplication and each variable has it's name for it's purpose.
9. Also the absolute value being declared earlier as a variable makes much more sense of how the computation flow works.

2.6.3 Member 3 : [redacted]

1. Java convention for naming was correct and no apparent mistakes found.
2. All variables declared belonged to a particular type and it was commented at the start for understanding.
3. Functions have a similar flow and do exactly what the function name entails
4. The azimuth, pitch, roll even if it was not apparent was clearly explained using the extensive comments for it.
5. Well commented code for every part, clearly understood purpose.
6. Comments for the UI elements were present in the MainActivity module as well as the xml resource module which is instantiating them.
7. Functions are to the point and not very long.
8. Why onAccuracyChanged was left empty was not apparent to me.

5

Let us see what were listed by member two the observations made by member two after going through the code the convention for proper variable and class naming is followed and the variables name when seen marks its purpose in other words clearly reveals their purposes. Azimuth role even if it was not apparent was clearly explained using the extensive comments for it that means the code was well documented.

Lines are commented well but sometimes too many comments for a single block reduces the code readability a little one observation critical of generation made by the evaluator. The ey elements are understood and I could perform a visualization from the buttons used from the comments. That is again a very interesting thing pointed out by the evaluator that even in the absence of actual execution of the code the manner in which the code was written makes it very clear to the evaluator how the display will look like once the code is executed that is very interesting observation.

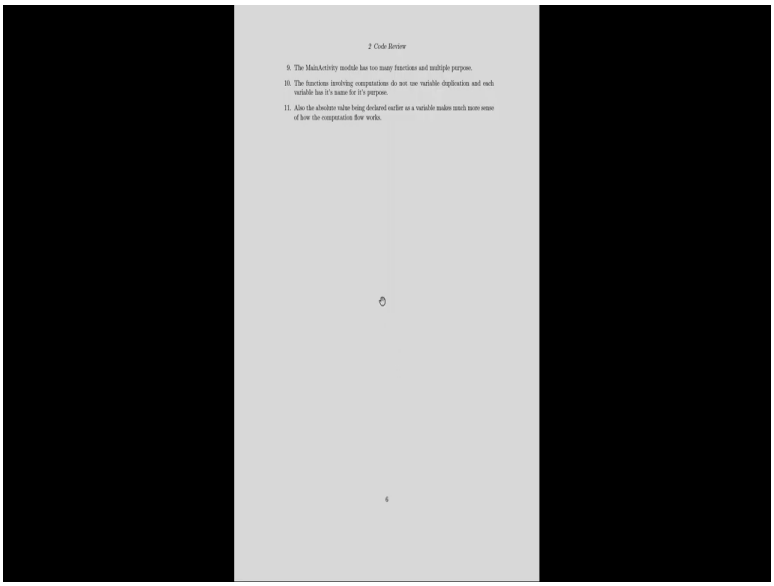
Functions are to the point and not very long the on accuracy change not apparent to me because it was an empty declaration was understood since it was an abstract class and there were comment lines present for it. The main module contains too many lines of code and split into different modules is recommended. The functions involving computations do not use variable duplication and each variable has its name for its purpose also.

The absolute value being declared earlier as a variable makes much more sense of how the computations flow works. These are the observations made by the second evaluator. Let us see the observations made by the third evaluator Java convention for naming was correct and no apparent mistakes found. All variables declared belong to a particular type and it was commented at the start for understanding.

Functions have a similar flow and do exactly what the function name entails the azimuth beach role even if it was not apparent was clearly explained using the extensive comments for it. While commented code for every part clearly understood purpose. Comments for the UI elements were present in the main activity module as well as the XML resource module which is instantiating them. Functions are to the point and not very long.

Why on accuracy changed was left empty was not apparent to me. So, all the evaluators appreciated few things mentioned that many things are in line with the conventions and standards but there are some points of concern.

(Refer Slide Time: 19:40)

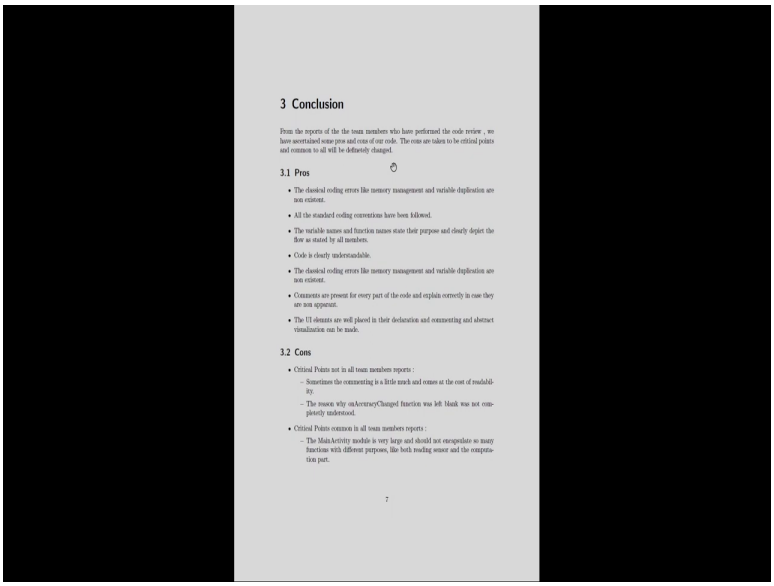


Few more comments made by the third evaluator where the main activity module has too many functions and multiple purpose. The functions involving computations do not use variable duplication and each variable has its name for its purpose. Also the absolute value being declared earlier as a variable makes much more sense of how the computation flow works. So, these are the observations made by the evaluator.

So, let us summarize the code for the system was developed and given to the team of evaluators having three evaluators. Each of them have gone through the code with the purpose of inspection based review in inspection based review execution is not required only line by line checking of code which at the beginning of the document is clearly mentioned that is a slow process. So, not more than 400 lines of codes are inspected at a time.

Otherwise the results may not be very good. After going through the codes each evaluator prepared some list of observations and those lists we have just seen for each of the evaluators.

(Refer Slide Time: 21:11)



3 Conclusion

From the reports of the the team members who have performed the code review , we have ascertained some pros and cons of our code. The cons are taken to be critical points and common to all will be definitely changed.

3.1 Pros

- The classical coding errors like memory management and variable duplication are non existent.
- All the standard coding conventions have been followed.
- The variable names and function names state their purpose and clearly depict the flow as stated by all members.
- Code is clearly understandable.
- The classical coding errors like memory management and variable duplication are non existent.
- Comments are present for every part of the code and explain correctly in case they are non apparent.
- The UI elements are well placed in their declaration and commenting and abstract visualization can be made.

3.2 Cons

- Critical Points not in all team members reports :
  - Sometimes the commenting is a little much and comes at the cost of readability.
  - The reason why onAccuracyChanged function was left blank was not completely understood.
- Critical Points common in all team members reports :
  - The MainActivity module is very large and should not encapsulate so many functions with different purposes, like both reading sensor and the computation part.

7

The last section of the document is the conclusion section. So, the observation list is provided to the developer before actually providing it to the developer the team evaluation team brainstorms with the list and comes up with a single list with all the critical observations listed. In this particular review document that list is mentioned in the conclusion section. From the reports of the team members who have performed the code review we have ascertained some pros and cons of our code.

The cons are taken to be critical points and common to all will be definitely changed that means Pros are the positive sides points of concerns and those need to be addressed before the code is for the tested. So, first the list of Pros the classical coding errors like memory management and variable duplications are non-existent. All the standard coding conventions have been followed the variable names and function names State their purpose and clearly depict the flow as stated by all members.

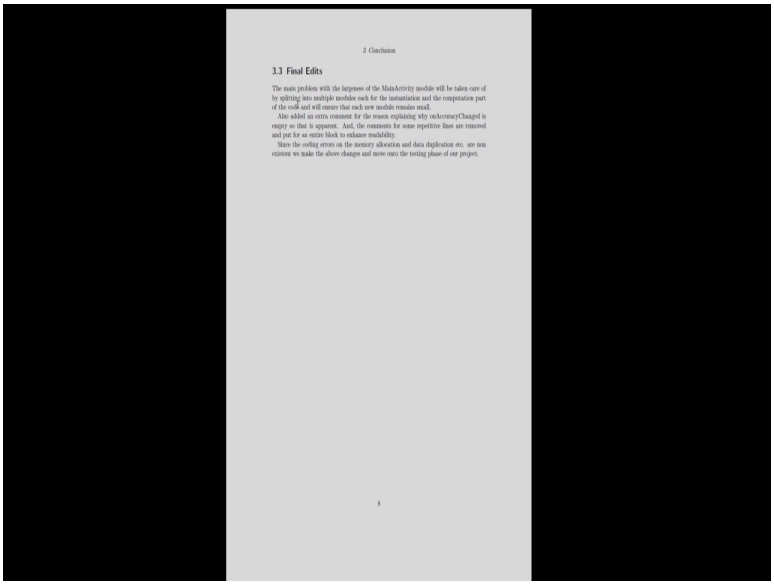
Code is clearly understandable the classical coding errors like memory management and variable duplications are non-existent. Comments are present for every part of the code and explained correctly in case they are non apparent. The UI elements are well placed in their declaration and commenting and Abstract visualization can be made these are the pros of the particular code developed by the development team.

The reports also reveal some cons once the reports are compiled those cons or the negative points are found out and those are listed here. Critical points not in all team members reports that means there is no agreement about these points sometimes the commenting is a little much and comes at the cost of readability. The reason why on accuracy changed function was left blank was not completely understood.

But there are some points which are reported by all the evaluators. Critical points common in all team members reports the main activity module is very large and should not encapsulate. So, many functions with different purposes. Like both reading sensor and the computation part. So, this is highlighted by all the evaluators whereas the other two points are not highlighted by all the evaluators.

So, based on this conclusion code needs to be refined ideally the common points are to be taken care of that means points that are found out from all the reports.

(Refer Slide Time: 24:20)



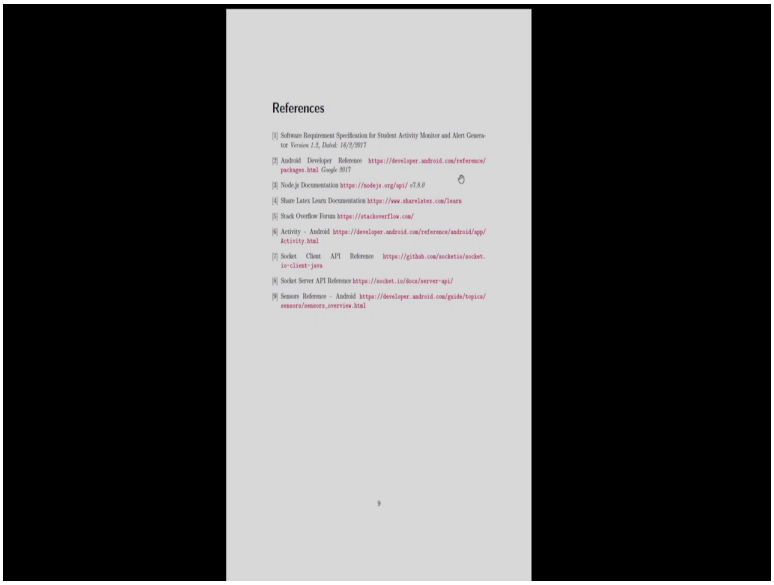
So, in the final edits part of the document it is mentioned that the main problem with the largeness of the main activity module will be taken care of by splitting into multiple modules each for the instantiation and the computation part of the code and we will ensure that each new module remains small it tells how to refine the code. Also added an extra comment for the reason explaining why on accuracy changed is empty.

So, that is apparent and the comments for some repetitive lines are removed and put for an entire block to enhance readability. So, in this exercise the comments that are not common to all evaluators are also taken care of. Since the coding errors on the memory allocation and data duplications etc are non-existence we make the above changes and move on to the testing phase of our project. So, here the development team declares that with the above actions taken based on the evaluator reports no further changes need to be made at this stage of code development.

And the code can be now formally tested with other testing methods. So, to summarize we talked about how ah code review document can be created how it looks like what should be there in the document and what actions are taken that also need to be mentioned in the document. These actions are primarily based on summarization of the reports that are received from the code evaluation team. Now there can be two types of issues one type of issue refers to those that are mentioned by individual evaluators.

But there is no agreement among all the evaluators about those issues. And the next type is the more crucial type where all evaluators agree that the issue exist or those issues exist. So, there is an agreement both these issues ideally should be taken care of in the code in particular the common issues are to be taken care of even if the other issues are not taken care off. However both ideally should be taken care off.

(Refer Slide Time: 27:09)



References

[1] Software Requirement Specification for Student Activity Monitor and Alert Generator Version 1.3, Dated: 16/2/2017

[2] Android Developer Reference https://developer.android.com/reference/packages.html Google 2017

[3] Node.js Documentation https://nodejs.org/api/ v7.8.0

[4] Share Latex Learn Documentation https://www.sharelatex.com/learn

[5] Stack Overflow Forum https://stackoverflow.com/

[6] Activity - Android https://developer.android.com/reference/android/app/Activity.html

[7] Socket Client API Reference https://github.com/socketio/socket.io-client-java

[8] Socket Server API Reference https://socket.io/docs/server-api/

[9] Sensors Reference - Android https://developer.android.com/guide/topics/sensors/sensors_overview.html

9

At the end of the document some references can be listed if while creating the document you have taken help from some sources those must be listed at the end. With that we have come to the end of this lecture this is continuation of lecture 26 where we discussed about code review and in this lecture we saw a case study on how to create a code review document I hope you understood the concepts and enjoyed the lecture looking forward to meet you all in the next lecture, thank you and goodbye.