Beware of human

Ady, László, Schuster, György PhD

Óbuda University
Kandó Kálmán Faculty of Electrical Engineering

2020. november 16.

Introduction

The software (some remarks)

More and more tasks are entrusted to our software.

These software perform a number of mission or safety critical tasks in many cases.

We have already proven in previous publications that a software cannot be fully tested.

Their complexity has already reached a the level that no one human being can understand and review.

Introduction

The software (some remarks)

More and more tasks are entrusted to our software.

These software perform a number of mission or safety critical tasks in many cases.

We have already pr software cannot be

Their complexity ha human being can u

The human

Human is an unreliable “machine” and as a result of this he or she will make mistakes, this is inevitable.

Human is also tiring, his attention digress and his or her work speed is limited.

The consequence of this is that the product may be unreliable upto some degree.

Introduction

The software (some remarks)

More and more tasks are entrusted to our software.

These software perform a number of mission or safety critical tasks in many cases.

We have already pr
software cannot be

Their complexity ha
human being can u

The human

Human is an unreliable “machine” and as a result of this he or she will make mistakes, this is inevitable.

Human is also tiring, his attention digress and his or her work

ıreliable

Then now what?

To avoid the above mentioned problems, developers use special tools.

In the most of cases, these tools are some kind of software or software-supported product.

However, we should not forget that these products have been developed with some level of human activity.

Life-cycle levels

Bundeswehr 1992

1. risk analysis, specification and contracting,
2. logical design,
3. physical design,
4. coding,
5. testing,
6. handover,
7. tracing.

Life cycle levels

Risk analysis, specification and contracting phase

For this phase, we can say that it is perhaps the most subjective step.
Unfortunately, engineers and technical staff have the least say at this stage. This, of course, is not just a software development problem, but is also true for each technical product.
From a technical point of view, the preparation of the specification is the most serious part at this stage.
Each of the next life cycle phases depends on this specification. Specification analysis is a very serious and difficult task. Therefore, it is advisable to use an aid in each case of development project.

Life cycle levels

Risk analysis, specification and contracting phase

For this phase, we can say that it is perhaps the most subjective step.
Unfortunately,
say at this stag
development p
product.
From a technic
specification is
Each of the ne
specification. S
difficult task. T
case of develo

Tools to help

High-level risk analysis can be aided by expert systems and - upto a certain extent - artificial intelligence applications.
These systems are able to estimate the given task, taking into account the experience of the recent period, the capabilities and resources of the given company. From a technical point of view, the preparation of the specification is the most serious part at this stage.
There are software products that can handle the specification and requirements, check and make it clear in some graphical way, for example by creating SysML diagrams and automatically generating and summarizing the descriptions of the given phases.

Life cycle levels

Logical design phase

In the logical design phase, the system is comprehensively designed based on the specification adopted in the previous phase.
The functional definition of the main components of the software and the definition of the overall requirements for the components take place, in this phase.

Life cycle levels

Logical design phase

In the logical design phase, the system is comprehensively
design
phase.
The fu
softwa
compo

Example:

let us consider the components of an aircraft's autopilot software as an example: heading, altitude hold, automatic engine control, navigation, etc. These consist of a number of additional parts, but a more detailed articulation of these is not necessary in this phase.

Life cycle levels

Logical design phase

In the logical design phase, the system is comprehensively designed based on the specification adopted in the previous phase.
The functional
software and t
components ta

Tools to help

To avoid problems due to human error, it is also necessary to apply a requirement management system that ensures step-by-step planning and generates the appropriate documentation.
In this case the SysML tools may be very useful.
We need to see clearly that these tools are not designed to replace people, they only facilitate a systematic workflow or “keep order”.
If we make a mistake during design, it will result wrong design. On the other hand, these tools ensure that one or more design steps are not left out of the development phase.

Life cycle levels

Physical design phase

In the logical design phase planned parts are designed here to the unit and algorithm level here.
The summary of the main requirements come from the previous level, which need to be detailes further.
It is extremely important not to miss any essential information eihter. A requirements management and documentation system is also required at this level.

Life cycle levels

Physical design phase

In the logical design phase planned parts are designed here to the unit and algorithm level here.

The su
previou
It is ext
eihter.
system

Example:

Take as an example the Mars Polar Explorer or the Mars Polar Lander, where one software module was calculated in SI units and the other software module in English units.

If in this case the communication between the two development teams had been proper, the errors in question would not have occurred and the loss of the two spacecraft would not have occurred.

If the developers had used proper requirements management, these unfortunate events would not have occurred.

Life cycle levels

Physical design phase

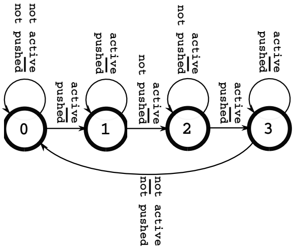
In the logical design phase planned parts are designed here to the unit and algorithm level here.
The summary of the main requirements come from the previous level, which need to be detailes further.
It is extremely important not to miss any essential information eihter. A requirements management and documentation system is also required at this level.

Tools to help

Software tools are also available for algorithm design, such as UML editors,

Life cycle levels

State graph



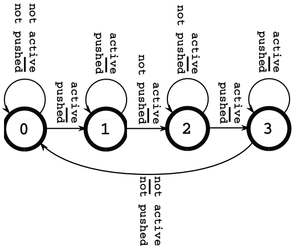
ırts are designed here

ːs come from the
es further.
ıy essential information
ınd documentation

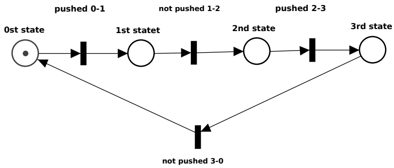
Software tools are also available for algorithm design, such as UML editors, state machine design software that supports dynamic behavior of the product

Life cycle levels

State graph



Petri net



rt

s

es

ny

nd documentation

Software tools are also available for algorithm design, such as UML editors, state machine design software that supports dynamic behavior of the product or Petri net designers etc..

Life cycle levels

Coding phase

In this phase the program and its operation data must be generated. The question is how we can do this most safely, too.

In many cases, we have the option of using a program generator partly or in full from previously produced plans or documents.

Life cycle levels

Coding phase

In this phase the program and its operation data must be genera

too.

In man

genera

docum

Generators

The advantage of using program generators is coding safety, but they are not the most economical in terms of both runtime and memory usage.

Further disadvantage may be that in the case of hidden program errors.

Therefore, in many cases, we use special generator, such as HALCoGens, for the coding, which generates functions corresponding to a graphically generated hardware configuration.

The "substantive" part of the program, on the other hand, is made with classical coding.

In this case programmers write the programs.

Life cycle levels

Coding phase

In this phase the program and its operation data must be genera too.

In man genera docum

Generators

The advantage of using program generators is coding safety, but they are not the most economical in terms of both runtime and memory usage.

Further disadvantage may be that in the case of hidden prog- ram er

Theref HALCo respon

The “s made v

In this

Programmers

It is important that the programming environment and the system used must be qualified. In this case, thre problems arise, all of which psychological:

- it can be a kind of fatigue and disinterest in those who do the actual work, caused by following several demands and prescription,
- most of programmers’ personality differ from average expectations,
- the zone state.

Life cycle levels

Coding phase

In this phase the program and its operation data must be
genera
too.

In man
genera
docum

Generators

The advantage of using program generators is coding safety, but they are not the most economical in terms of both runtime and memory usage.

Further disadvantage may be that in the case of hidden prog-
ram er
Therefc
HALCo

Programmers

It is important that the programming environment and the system used must be qualified. In this case, thre problems arise,

those who
/eral

n average

To avoid them:

- applied restrictions, such as the MISRA, set of rules need to be introduced.
- compliance monitoring can be automated,
- pair programming,
- compliance with mandatory breaks.

Life cycle levels

Testing phase

In static testing, human work is essential. There are attempts at artificial intelligence applications, but for now, they are in their infancy in safety-critical areas.

In the case of static testing application of human force has psychological characteristics, which also play a significant role.

In dynamic testing, automatic testing is a much more applicable procedure. In the case where a large number of test cases need to be performed in a short time, there is no other option.

Life cycle levels

Testing phase

In stati
at artifi
their in

In the c
psycho
role.

In dyna
applica
test ca
other c

Static testing

In static testing, human work is essential. There are attempts at artificial intelligence applications, but for now, they are in their infancy in safety-critical areas.

In the case of static testing application of human force has psychological characteristics, which also play a significant role, such as:

- fatigue,
- negligence ”...well, so far there has been no mistake, there will be no more...”,
- lack of time.

Life cycle levels

Testing phase

In stati
at artifi
their in

In the c
psycho
role.

In dyna
applica
test ca
other o

Static testing

In static testing, human work is essential. There are attempts at artificial intelligence applications, but for now, they are in their infancy in safety-critical areas.

In the
psycho
such a

- fa
- n
  th
- lack of time.

Dynamic testing

In dynamic testing, automatic testing is a much more applicable procedure. In the case where a large number of test cases need to be performed in a short time, there is no other option.

The question is: who tested the test program?

Life cycle levels

Testing phase

In stati
at artifi
their in

In the
psycho
role.

In dyna
applica

Static testing

In static testing, human work is essential. There are attempts at artificial intelligence applications, but for now, they are in their infancy in safety-critical areas.

In the
psycho
such a

Dynamic testing

In dynamic testing, automatic testing is a much more applicable procedure. In the case where a large number of test cases

other option.

To avoid problems:

- checklists and appropriate methods created to be applied,
- job rotation,
- using code review,
- giving enough and sufficient testing time,
- testing the test software.

Life cycle levels

Handover phase

In case of safety-critical systems, the delivery of the completed software is much more important than in case of a general-purpose system.

Assuming that in the case of such systems, the ordering specialist is able to plan the handover process in advance. This phase is based on an initial specification and set of requirements.

Based on the documents produced by the requirements and the acceptance test plan is also completed in the first and second phases, largely automatically. However, the handover phase is typically based on human activity.

Life cycle levels

Handover phase

In case of safety-critical systems, the delivery of the
comple
genera

Assum
special
This ph
require

Based
the acc
second
phase

Static testing

A psychological problem is the composition of the transferring team and the receiving team. In this case, it is very important to exclude excessive emotional factors and to put professionalism to the maximum level.

This phase is based on an initial specification and set of requirements.

Based on the documents produced by the requirements and the acceptance test plan is also completed in the first and second phases, largely automatically.

Life cycle levels

Handover phase

In case of safety-critical systems, the delivery of the
comple
genera

Assum
special
This ph
require

Based
the acc
second
phase

Static testing

A psychological problem is the composition of the transferring team and the receiving team. In this case, it is very important to excl
nalism

This ph
iremen

Based
the acc
cond phases, largely automatically.

A terrible problem:

A problem that does not necessarily arise when handing over safety-critical systems is the retrospective requirement statement.

In our experience, this is an increasingly common problem that sooner or later it also occurs in this environment.

Life cycle levels

Handover phase

In case of safety-critical systems, the delivery of the
comple
genera

Assum
specia
This ph
require

Based

Static testing

A psychological problem is the composition of the transferring team and the receiving team. In this case, it is very important to excl
nalism

This ph
iremen

A terrible problem:

A problem that does not necessarily arise when handing over safety-critical systems is the retrospective requirement statement.

problem that

To avoid problems:

It is important that the customer can inspect any documentation to ensure that all phases have been performed in accordance with the regulations.

You can also use an external expert for this. If required by the system, the relevant authorities are also involved in the acceptance process and are responsible for the authorization issue.

All steps of the acceptance process should be recorded step by step and the appropriate documents mast be produced.

Life cycle levels

Follow-up phase

In many cases, a software can be found to contain some hidden errors. Many times this is revealed in such a way that the error is not appeared in the case of a previous customers.

This is typically the case of component reuse when we involve components that have previously been manufactured and tested in development.

We tested it in vain, but we may still find a ”bug” in another product or during a new development in the testing phase.

Life cycle levels

Follow-up phase

In many cases, a software can be found to contain some
hidden
the err

This is
involve
and tes

We tes
produc

Experiences:

This is the case when the psychological effect reappears, some of the manufacturers are trying to get rid of the error correction, the reasons for which may be as follows:

- in the case of a delivered software, the cost of the repair is at least 50 times higher than repair during production,
- try to hide the problem, saying there has been no problem so far, then there will be no more,
- try to hide the problem in order to maintain a favorable professional image of themselves so that they do not make a mistake.

Life cycle levels

Follow-up phase

In many cases, a software can be found to contain some hidden the err

This is involve and tes

We tes produc

Experiences:

This is the case when the psychological effect reappears, some of the manufacturers are trying to get rid of the error correction, the reasons for which may be as follows:

- ir is
- tr p
- tr p m

One not-too-brief note:

We do a research on software reliability, where we look at the company, not the software itself.

This research requires statistics on failures of the software produced by the given company.

We cannot get any information. The basis of reference is that these values are internal secret data, so we cannot obtain them.

Life cycle levels

Summary

1. Human is an unreliable “machine”.

Life cycle levels

Summary

1. Human is an unreliable “machine”.
2. There are a number of automation options.

Life cycle levels

Summary

1. Human is an unreliable “machine”.
2. There are a number of automation options.
3. These tools are not reliable enough in safetycritical cases.

Life cycle levels

Summary

1. Human is an unreliable “machine”.
2. There are a number of automation options.
3. These tools are not reliable enough in safetycritical cases.
4. Human work is inevitable.

Life cycle levels

Summary

1. Human is an unreliable “machine”.
2. There are a number of automation options.
3. These tools are not reliable enough in safetycritical cases.
4. Human work is inevitable.

Conclusion: if we take into account the problems of psychology and their solution, the well-interpreted and correct use of aids, successful projects can be carried out.

Thank you for your kind attention!