HUMAN ERROR IN THE SOFTWARE GENERATION PROCESS

Trevor Cockram Rolls-Royce plc
Jim Salter and Keith Mitchell Lloyd's Register
Judith Cooper and Brian Kinch Lucas Engineering & Systems
John May Open University Dept. Computing & Nuclear Electric

Summary

In this paper we discuss how faults are introduced into the software generation process. The Fault Analysis of the Software Generation Process (FASGEP) project has classified faults into Random and Symptomatic. Symptomatic faults are those faults where the input to the process was correct; but the output from the process was incorrect due to an error in the process. Random faults are those faults for which no specific cause for the fault can be identified, This paper discusses the nature of random faults and to what extent they can be attributed to human error.

Software process decomposition shows that the human engineering process can be described as a combination of intellectual (novel) processes and more mechanistic processes. The mechanistic processes, e.g. the use of tools, can be considered to be a specific form of human-computer interaction via a particular human-machine interface.

The classification of human error within the FASGEP project takes into account the work of Rasmussen, specifically the classification of Rule-Based errors, Skill-Based errors, and Knowledge-Based errors. The causal relationships for each of the classifications have been developed into a causal network. A type of graphical probability model is based on this.

A probability model, of which GPMs form a part has been generated to determine the susceptibility of a software generation process to the introduction of faults. The model uses evidence from metrics collected on team, management, environment and communication attributes.

1. Introduction

Software has been developed for use in many safety critical and safety related applications. In using software for these applications it is necessary that we have understanding and control of the processes used to generate this software. There are many standards, methods and tools available to support the generation of software. Certain standards include requirements for personnel involved in the generation of software e.g. [1], but these are limited to formal qualifications and required experience. This paper however, addresses the human involvement in the software generation process and more particularly the errors introduced by humans into software.

In this paper we discuss how faults are introduced into the software generation process. We have defined the scope of the software generation process for the purposes of this paper as the stages of development between receiving a software requirements specification through to and including coding, i.e. essentially the design activities. We consider the involvement of people with the process and the type of thought processes used by the persons involved.

In particular we consider a model of the fault generation process. This model was developed as part of the Fault Analysis of the Software Generation Process (FASGEP) project which is being carried out within the DTI/IED Safety Critical Systems Programme.

2. Classical Theory and how it relates to the FASGEP Model

Reason [2] indicates that human error is intimately related to the concept of "intent". Error is only a meaningful term when applied to intended (planned) actions that fail to achieve the desired goal without the intervention of some chance or unforeseeable agency. Thus, non-intentional, involuntary and spontaneous actions are not errors. Reason identifies two basic types of error: slips and mistakes.

Slips are where actions do not go according to plan e.g. slips of the tongue, slips of the pen, slips of action. Mistakes occur when the plan itself is inadequate to achieve its objectives. A further category, lapses, are a form of slip, essentially involving a failure of memory which is not necessarily revealed in actual behaviour and may only be apparent to the individual concerned.

Reason's error types can be related to the stages in the cognitive process at which they occur. For the cognitive stages of planning, storage and execution the primary error types are mistakes, lapses and slips respectively.

A further relationship exists between these error types and Rasmussen's [3] classic model of skill, rule and knowledge-based behaviour. Slips and lapses tend to occur at the skill-based level, whereas mistakes occur at the rule-based and knowledge-based levels. Rule-based mistakes are primarily due to misapplied expertise, where some pre-established plan or problem solution is applied inappropriately. Knowledge-based mistakes generally occur due to a lack of expertise, where no off-the-shelf solution exists and an individual is forced to work out a plan of action from first principles.

Reason goes further and highlights the likely failure modes at each level of behaviour. For example skill-based errors occur due to control-mode failures of both inattention and over-attention. Rule-based errors can arise from the misapplication of rules or the application of incorrect rules. Knowledge-based errors can arise due to selectivity, biased reviewing and a number of other factors.

The FASGEP project considers each of the three levels of human behaviour and attempts to define the most significant attributes of the software development process (and metrics for measuring each attribute quantitatively) which influence performance at the appropriate level and therefore lead to errors and the introduction of faults in the software product. These faults may or may not be recovered by review processes within the development life-cycle [4].

It is worth noting, here, that this categorisation and application of attributes relates primarily to Reason's behavioural and contextual levels of human error classification (i.e. it indicates the factors likely to lead to errors of a particular type (slips, lapses or mistakes)), but does not address the conceptual level of error classification, which is concerned with cognitive mechanisms involved in error production.

Reason himself distinguishes between error type and error form. Error forms are recurrent varieties of human fallibility (what others have called psychological error mechanisms) that appear in all kinds of cognitive activity, irrespective of error type.

Thus, the FASGEP approach cannot and does not identify attributes of the software development and review processes which influence human performance in psychological terms (e.g. specific cognitive or socio-psychological factors leading to an increased tendency for focusing and mindset, fixation, overconfidence or other problems giving rise to judgmental and decision errors.) The attributes and associated metrics do, however, identify significant factors in the process which influence the introduction of faults in the software, due to human error. These attributes represent an assimilation of current thinking in recent attempts to obtain measures of fault introduction in software, and are the best judgements of the members of the FASGEP project consortium.

3. Fault Introduction Model

The FASGEP analysis model consists of two parts, an inner model to determine the probability of the number of faults generated or removed by a particular atomic process, and an outer model which propagates the results from the inner models as the project progresses through the software development life cycle and results in a probability distribution for the estimated number of faults remaining at the end of the development life cycle [5].

The FASGEP project has identified two classes of graphical probability models (GPM) for the inner (or fault introduction) model:

1) The development process which introduces faults
2) The review process which identifies faults which in the semantics of the FASGEP project is responsible for fault removal.

The purpose of the development fault introduction models in each atomic process is to calculate the probability of faults being introduced during that atomic process. This probability is in the form of a distribution over fault numbers and known as fault propensity. At the highest level the total fault propensity is calculated from a convolution of the fault propensity for Symptomatic faults with the fault propensity for Random faults, but the GPMs are also used to perform completely general Bayesian updating calculations. The GPMs enable fault propensity to be calculated from a fusion of the prior probability distribution for each net, and evidence data collected from metrics of the generic atomic process attributes which are observed for each atomic process.

It is the development fault introduction model which assess the possibilities for human errors. A GPM allows a collection of factors, and the probabilistic relations between them, to be modelled. The structure of the networks are shown below, on which a GPM is based describes which variables influence each other, and conditional probability matrices are used to weight the various influences.

Symptomatic faults are those faults which have been identified to be caused by a defect in the process, rather than the people carrying out the task. It may be that different atomic processes by their nature will exhibit different failure characteristics; the project has been careful to maintain a generic model of software development, without specifying a particular process or life cycle. The factors influencing symptomatic faults were identified in the process modelling activity as: Goodness of (process) interfaces which includes undefined processes, Project Management Quality, Quality of Input Product and Goodness of Method.
Random Faults in the context of FASGEP have been identified as being associated with faults introduced by human behaviour.

3.1 Skill-Based Errors

Skill-based errors are identified by problems due to inattention or over attention to the specific task. These are typically identified by slips, omissions or repetitions in the product being produced. The causal net for skill-based errors uses attributes of ability, motivation and environment quality (see figure 1).

The measurement of a person's intrinsic ability to carry to a specific task is difficult to determine. One method is the use of psychometric tests e.g. [6]; however, this method was not considered appropriate for the initial case studies in the FASGEP project and a subjective estimate of ability was used.

Job satisfaction, morale and workload match, contribute to self motivation and are closely inter-linked and it would be difficult to recognise a situation in which all three factors are not equally important. All three factors can vary: with time; between individuals; and as a team. It is recognised that high levels of job satisfaction and morale can result in high levels of self motivation, but the relationship is very complex and can be influenced by other personal factors.

High levels of job satisfaction are generally only achieved when the basic needs of security and salary are satisfied. There are occupations where salaries are low but job satisfaction is high, but the norm is the reverse. Low salaries and job insecurity can be a source of stress factors, which have a detrimental effect on overall job satisfaction. High salaries and job security alone may not be enough to produce high job satisfaction. Other more complex emotional factors are required for high levels of job satisfaction.

Job satisfaction and team interaction are significant factors in determining morale in the working environment. Recognition of skills, and of being a valued member of a team, also contributes to high morale. Morale can be adversely affected by lack of recognition, inability to communicate errors freely, isolation and general lack of cohesion within the peer group. If these negative factors are present in other team members, team morale and motivation would be expected to be low.

Working Environment Quality is a complex problem to measure. A separate causal net was developed for this quality. The working environment quality is influenced by many attributes: by the Workstation Quality, Effectiveness of Communication, Working Environment Satisfaction and by the presence (or absence) of individual control over the working environment.

The Workstation Quality of the individual team member's own working area (e.g. their allocated desk or wherever they spend most of their time) has several attributes: the comfort of the individual is felt to be the overriding factor.

The comfort factor assesses the quality of the ambient environment. In this assessment it was felt that the following attributes were considered important: lighting, heating, noise and ventilation.

The European Commission [7] have set down statutory requirements for persons working on computer terminals. These requirements are reflected in the quality of facilities factor.

The team members' satisfaction with their working environment entails both the level of distraction to which they were subjected and the quality of the services provided. Distractions are known to have a severe detrimental effect on the ability of an individual to perform a given task accurately. The level of distraction is found from attributes associated with the number of neighbours not contributing to the task in hand and the number of neighbours in total. Noisy items, especially intermittently noisy items, contribute to the level of distraction In determining the quality of the services provided to the team members, the availability of services is obviously the overriding factor. The secondary factor is considered to be the quality of the service provided and finally the proximity of the said services.

3.2 Rule-Based Errors

Rule-based errors occur because of the working method and procedures imposed. Poor working methods and procedures can give rise to disagreement, disillusionment or even resentment from the workforce. This may be because the rules are: inelegant, inadvisable, too strong, or too general. It could be argued that a professional engineer would not make this type of error but, even so, there may be problems with interpretation, or implementation, that would lead to this type of error being produced. (The causal net is shown in figure 2)

Task appropriateness and the quality of project management have been considered to be the main attributes for rule based errors.

A task can be regarded as appropriate when an individual or a team have the necessary skills to complete the task satisfactorily and the task has been adequately defined. Inappropriate tasks may be recognised by the level of errors that appear in the task life cycle. Experience-level and task-experience matching are seen as the most significant factors in task appropriateness.

Four factors have been highlighted as having significant influence on project management quality. Leadership is a contributor to good quality management since it can demonstrate commitment to a project from the top of the organisation. Team quality (continuity, cohesion and team size) and effective communications are also important.

Team attributes are those characteristics that produce co-operation, camaraderie and good internal communications which tend to lead to a "quality" team output. A good team has a capability level greater than that achievable by the efforts of the team members acting as individuals. Team cohesion and continuity have a more significant effect than team size.

3.3 Knowledge Based Errors.

Rasmussen [8] has indicated that experience is the most important of the attributes in determining the number of knowledge-based errors. A good team has been shown [9] to result in a reduction in the number of errors introduced. Shared experience has therefore been considered to have the highest weight in assigning the probabilities in this distribution (see Figure 3).

Task comprehension has been determined from the questions on task familiarity and an adequate task description. The experience attributes are obtained from the responses to multiple questions on experience. The shared experience factor determines the effect of the team working together to pool knowledge in solving problems. An individual working as part of a cohesive team can use the experience of others, and conversely the lack of cohesion within a team prevents experience being shared [9] [10]. Cohesion in this context has been determined from social interaction, team dedication and the number of new members joining the team .

3.4 Communication

Communication is an important human factor in the generation of software errors, and its effects can be seen in both random and symptomatic types of faults. In the FASGEP project we have developed a causal net for the effectiveness of communications which is used in both random and symptomatic nets.

The effectiveness of the communication attribute varies according to the complexity rating of the hierarchy. For simple hierarchical structures, informal communications are considered more appropriate than formal ones since the team is likely to be small with little problem in discussing issues with their peers and superiors. Complex hierarchical structures however, require that communications be more formal to ensure that the correct people are kept informed of all relevant (and only relevant) developments.
To assess the complexity of the reporting hierarchy, data is captured detailing the number of reporting levels and the number of sites involved in the development. During the development of the Causal Network, it was realised that a further measure of complexity is to be gained from capturing the number of lateral paths in the reporting structure.

In determining the Effectiveness of Formal Communication, it was decided that the flexibility of the communication within the team was the best indicator of effective communication, the type and frequency of team meetings were considered to be of similar importance, and the quality of feedback to team members was considered to be a relatively minor indication of effective formal communication.

The main indicator of effective informal communications was decided to be the quality and quantity of verbal communication in preference to non-verbal communication.

4. Data Collection

It has been shown many times that the collection and storage of data is a vital aspect of any unification framework. Poor data-collection techniques and requirement definitions have been the causes of the limited success and acceptance of several other projects aimed at developing predictive models for software development.

Although the general requirements for a successful data-collection exercise include automating as much of the process as possible using formal measures, it was realised early in the FASGEP project that human factors were a large contributor to the introduction of faults in software. These are expected to include factors such as the individual's job satisfaction and morale which can realistically only be determined by direct questioning of the individual. This requires the use of a questionnaire approach.

Additionally, since FASGEP requires the collection of data from several sources, each of which uses different development processes and fault collection techniques, it was decided that to collect such data by automatic means was at this stage impractical and would impose severe restrictions on the data available to the project. Thus, the questionnaire approach was deemed the most appropriate technique at this stage of development of the predictive model.

Questionnaires have several inherent weaknesses that the FASGEP project recognises and has attempted to minimise by careful design:

(i) Question wording directly affects the validity and reliability of a questionnaire [11].

(ii) The format of the questions is also important: should they be open or closed. Closed questions can force inappropriate response, but are easier to capture and check.
(iii) Since questionnaires rely solely on the interpretation and feelings of the respondent the answers may be biased and may exhibit some degree of subjectivity.

(iv) Respondents are sensitive to the context in which the question is asked, as well as the particular words used to ask it. As a result, the meaning of almost any question can be altered by a preceding question [12].

As already stated, it was an obvious requirement that the project be able to collect data on the feelings and opinions of the individuals involved in the software development. In cases such as these, the approach of using a questionnaire actually becomes beneficial with respect to the normal problem areas of subjectivity and bias. It is these aspects of human emotions that we are interested in capturing and using as evidence of the motivation and satisfaction of the individual.

5. Case Studies

The aim of the FASGEP case studies is "to provide vehicles for testing the FASGEP Model and Method as they evolve throughout the project". That is, they will be used for calibration [1] and verification [2]. Three types of case study projects are being used by the FASGEP project.

(i) Past Projects — Projects that have been completed and delivered to the customer. Although the fault data can be considered to be complete, the collection of the human factors data is difficult since the development team is likely to have spilt up. Also much of the other data required by FASGEP is unlikely to have been captured during the development.

(ii) Ongoing Projects — These are projects in which the development is already some way to completion. Although some data may have been lost, these are useful to observe the effect of introducing new collection requirements on the team both in terms of attitude and resource requirements.

(iii) New Projects — New projects are those for which the FASGEP data collection scheme is implemented from the very start of the development. Thus all data is available. These allow the identification of the changes in the team throughout the development and how these changes can affect the integrity of the final software.

6. Results

[1] Calibration: the process by which the parameters of the model are adjusted to provide a result that fits observed data accurately.

[2] Verification: the process of ensuring that the model works correctly

At the time of writing (August 93) none of the case studies of the type described above have completed the data-collection exercise. However some initial analysis of the fault reports in two case studies were available; these are shown below:

Test Case A

Application: Aerospace - Fuel Control System
Language: Proprietary (Assembler)
Lifecycle: Structured - with incremental delivery
Sample size: 45 errors - of which 11 Symptomatic; 27 Random; 1 unclassified; 7 not faults

Test Case B

Application: Aerospace - Integrity Monitoring System
Language: Ada
Lifecycle: Evolutionary
Sample size: 28 errors - of which 12 Symptomatic; 14 Random; 2 not faults

The random faults were further analysed to estimate the cognitive level of the error: Skill based = 2; Rule based = 2; Knowledge based = 9; there was one fault where there was insufficient information to categorise.

It is hoped that by the time the paper is presented, the results from several case studies will be completed and these will be reported.

7. Conclusions

To improve the software generation process, it is clear that account must be taken of the humans involved in the process. This is in addition to the tools, methods and procedures used. The FASGEP method provides a means of estimating the number of detectable faults in a product caused by human error using the objective and subjective information available.

A potential use for this method could be for testing scenarios before changes are made to the development teams and working environments are made.
Further work is now required in validating the method using case studies.

8. Acknowledgements

The authors gratefully acknowledge the partial funding provided by DTI under the Safety Critical Systems programme reference IED/1/9004.

The FASGEP consortium consists of Lloyd's Register of Shipping, Lucas Electronics, Lucas Engineering & Systems, Nuclear Electric, The Open University and Rolls-Royce.

The authors would also like to thank Sarah Grey, Dr Lesley Winsborrow, Steve Connolly, Martin Cottam, Reg Parker, David Nicoll, Jon Speer, Nick Bird, Dr Eddie Williams and Dr Mike Falla for their considerable contribution to the project

References

[1] Defence Standard 00-55 interim issue 1 April 1991

[2] Reason J. Human Error. Cambridge University Press Cambridge, 1990

[3] Rasmussen J. Skills, rules, knowledge: signals, signs and symbols and other distinctions in human performance models. IEEE Trans Systems, Man and Cybernetics SMC-13 (1983), 257-267

[4] May J, Hall P, Zhu H, Cockram TJ, Bird N, Fault Prediction for the Software Development Process. Proceedings of IMA conference on the Mathematics of Dependable Systems, Egham, Surrey September 1993

[5] Hall P et al., Integrity Prediction during Software Development Proceeding of Safecomp'92 conference, Zurich October 1992

[6] Aiken LR, Psychological Testing and Assessment, Allyn & Bacon 1988

[7] European directive 90/270/EEC 29 May 1990

[8] Rasmussen J. Information Processing and Human Machine Interaction North Holland, 1986

[9] DeMarco T, Lister T. Peopleware. Productive Projects & Teams Dorset House 1987

[10] Basili V, Reiter R. An investigation of human factors in software development Computer Dec 1979: 21-38

[11] Ed. J Richardson. Usability Evaluation, RACE project deliverable (ISSUE programme) Nov 1992

[12] Converse JM, Presser S. Survey Questions - Handcrafting the standardised questionnaire Sage Publications, 1986