# Methodology of functionality selection for water management software and examples of its application

K. N. Vasilyev

## ABSTRACT

When developing new software products and adapting existing software, project leaders have to decide which functionalities to keep, adapt or develop. They have to consider that the cost of making errors during the specification phase is extremely high. In this paper a formalised approach is proposed that considers the main criteria for selecting new software functions. The application of this approach minimises the chances of making errors in selecting the functions to apply. Based on the work on software development and support projects in the area of water resources and flood damage evaluation in economic terms at CH2M HILL (the developers of the flood modelling package ISIS), the author has defined seven criteria for selecting functions to be included in a software product. The approach is based on the evaluation of the relative significance of the functions to be included into the software product. Evaluation is achieved by considering each criterion and the weighting coefficients of each criterion in turn and applying the method of normalisation. This paper includes a description of this new approach and examples of its application in the development of new software products in the are of the water resources management.

**Key words** | formalisation approach to the selection of functions, river and floodplain modelling software

**K. N. Vasilyev**
CH2M HILL,
Burderop Park,
Swindon,
SN4 0QD,
UK
E-mail: *kostyavasiliev@hotmail.com*; *Kostya.Vasilyev@ch2m.com*

## INTRODUCTION

Over the last 20–30 years, programs automating calculations for flood damage evaluation (ISIS, MIKE 11, HEC-RAS, etc.) have appeared on the market. In the last few decades, advances in computer science and Internet technologies have accelerated the automation of such calculations. There have been substantial improvements in how results are presented. Developers are now providing licences for the software, and expertise in river model development and processor time as services.

Such services promise to be highly relevant for many client organisations (government agencies, local councils, etc.) due to their lower cost. From the perspective of the developers, this trend of increased automation is likely to lead to an expanded client base. The developers will, in particular, aim to further implement the new Internet technologies that enable the modelling software and services to be provided via cloud computing (software as a service).

When developing new software products, and adapting existing software, project leaders have to decide which functionality to keep, adapt or develop. Depending on the complexity of the problem, when selecting new functions, it is recommended that various methods of decision-making are applied, taking a formalised approach to the task. Here one has to consider that the cost of errors in the specification phase is extremely high.

In this paper a formalised approach for selecting functions for new software products is proposed, that takes into account the basic criteria for the selection of functionality. Applying this approach to software development will allow the number of errors made during the specification phase to be minimised.

## METHOD

This section describes the criteria for selecting which functions to include in a new software product and the main steps used to prioritise the selection.

doi: 10.2166/wst.2013.500

## Criteria for the selection of functions for the inclusion into a new software product

When working on the development and support of software products in the area of water resources management and evaluation of the economic damage caused by flooding, the author has identified the following criteria as important when choosing which functionalities to include in the software:

- Compliance of the function with the purpose of the software usage
- Technical complexity of the creation of the programming code
- Function's contribution to the competitive appeal of the software product
- Costs involved in developing the programming code for the function
- Potential problems related to supporting the product caused by using of the function
- Risk of exceeding the budget when developing programming code for the function
- Risk of not meeting the deadline for the release of the software.

## Diverse criteria method for the selection of functions for inclusion in a new software product

When making decisions regarding which functions to select, a model based on the evaluation of the relative importance of the function can be applied. This evaluation is done by applying a rationing method using several criteria and taking their weights into account (Volkova & Denisov 2010). Shown below is the matrix of compliance of the functions and criteria used during their evaluation, in the form of a table. The table uses the following designations: $\gamma_x$ – weight coefficients of the criteria, $w_{jx}$ – relative weight coefficient of the function being evaluated according to certain criteria, coefficient of relative importance of the function j according to the criterion i; A, B, …, N – functions (see Table 1).

The project manager first needs to input the values of the weights of the criteria and then enter the values of the relative weights of each function according to the proposed criteria detailed in Table 1. The determination of the relative weights of each function can be delegated to a different member of staff in the department, who is also involved in the project. In addition, a poll of all key staff (an 'expert group') and subsequent comparison of the results are

**Table 1** | Evaluation of relative weights of the functions proposed for a new software product

| **Criterion** | | **Compliance of the function with the purpose of the software usage** | **Technical complexity of the function** | **Function's contribution to the competitive appeal of the software** | **Costs involved in developing the programming code for this function** | **Potential problems related to supporting the software due to the inclusion of this function** | **Risks of exceeding the budget when implementing this function** | **Risk of not meeting the deadline of the next release of the software** | **Estimates of significance of the functions** |
|---|---|---|---|---|---|---|---|---|---|
| Weights of the criteria | | $\gamma_1$ | $\gamma_2$ | $\gamma_3$ | $\gamma_4$ | $\gamma_5$ | $\gamma_6$ | $\gamma_7$ | |
| Relative weights of functions (A–N) | A | $w_{a1}$ | $w_{a2}$ | $w_{a3}$ | $w_{a4}$ | $w_{a5}$ | $w_{a6}$ | $w_{a7}$ | $r_{ia}$ |
| | B | $w_{b1}$ | $w_{b2}$ | $w_{b3}$ | $w_{b4}$ | $w_{b5}$ | $w_{b6}$ | $w_{b7}$ | $r_{ib}$ |
| | … | | | | | | | | |
| | N | $w_{n1}$ | $w_{n2}$ | $w_{n3}$ | $w_{n4}$ | $w_{n5}$ | $w_{n6}$ | $w_{n7}$ | $r_{in}$ |

possible. In order to increase the objectiveness of such estimates, the members of staff, whose estimates can be very different from each other, may be given the option to substantiate their estimates. As a result of this discussion, some members of staff may rethink their estimates or other points may be discovered, which may require a separate discussion. This process may increase the effectiveness of the work on the functionalities and the quality of the product being created. Obviously, when doing a poll of several members of staff, this model may be used in several rounds, i.e. a Delphi procedure can be used (Linstone 2002).

For coherent estimates for each of the functions during the input of the values of the relative weights of each function, the rules listed in Table 2 should be followed. When filling in the table, one has to keep in mind that the weight coefficients of the criteria and the relative weight coefficients are normalised:

$$\sum_{x=1}^{m} \gamma_x = 1 \quad \text{and} \quad \sum_{j=1}^{n} w_{jx} = 1 \tag{1}$$

After filling in the table, it is necessary to calculate the estimates (values of relative significance) of each function taking into account the weight coefficients of the criteria:

$$r_{\text{ij}} = \sum_{x=1}^{m} \gamma_x * w_{jx} \tag{2}$$

In addition, the condition of normalisation is also checked:

$$\sum r_{ij} = 1 \tag{3}$$

When following the rules in Table 2, the estimates $r_{ij}$ allow one to select functions which best fit the project: those with the lowest cost of development of the code, which comply with the purposes of the product etc. based on the opinion of the expert group (and taking into account the weight coefficients, defined by the project manager/expert group).

## APPLICATION

In this section two examples, where the diverse criteria method could be applied, are described. The method is first used to analyse the decision-making process, which took place when developing ISIS FAST software. Then the method is applied to analyse the decision-making process for the inclusion of the new functionalities into ISIS FAST software. The values of the relative weights of the criteria are based on the opinions of the author and another member of the software team. These values were specified after the pieces of software or functions had already been decided, although they coincide with the decisions made using a less formalised approach. Brief descriptions of the ISIS FAST software and the ISIS suite of software are also given in the section.

### ISIS Suite of Software

The ISIS Software Suite (www.ch2mhill.com/isis) is a flexible range of tools for designing engineering schemes, planning cost-effective flood mitigation measures and developing catchment strategies. At the heart of the system is ISIS 1D, a full hydrodynamic simulator for modelling

**Table 2** | The rules of coherence of weights of functions

| No | Criterion | Rule of coherence | Symbolic representation of coherence |
|---|---|---|---|
| 1 | Compliance of the function with the purpose of the software | The larger the compliance, the bigger the estimates | ↑ |
| 2 | Technical complexity of the code | The more complex the function, the smaller the estimates | ↓ |
| 3 | Function's contribution to the competitive appeal of the software product | The greater the contribution, the bigger the estimates | ↑ |
| 4 | Cost of code development | The bigger the cost, the smaller the estimates | ↓ |
| 5 | Potential problems with supporting the product | The larger the problems, the smaller the estimates | ↓ |
| 6 | Risk of exceeding the budget | The bigger the risk, the smaller the estimates | ↓ |
| 7 | Risk of not meeting the release deadline | The bigger the risk, the smaller the estimates | ↓ |

flows and levels in open channels and estuaries. ISIS 1D was developed in the 1970s by Halcrow, now CH2M HILL company. Since then the software has been constantly evolving. New tools such as ISIS MAPPER ISIS 2D and others were created, which together with ISIS 1D make ISIS Software Suite. One of the recent additions to this range of tools is ISIS FAST, which was released in April 2011.

## ISIS FAST software

The ISIS FAST software allows modellers to calculate the water levels in floodplains in a short amount of time, but retain sufficient accuracy for specific use. This is especially useful for flood forecasting in real time, probabilistic analysis and national-scale flood mapping (Wicks *et al.* 2012). The time it takes for ISIS FAST to calculate water levels is measured in minutes, whereas the similar runtime measurement for 'traditional' two-dimensional programmes (TUFLOW or ISIS 2D) is hours or days, although the latter programmes may define the levels with higher precision (ISIS Manual 2012).

## Decision-making process for the development and release of ISIS FAST

Prior to the start of the development of ISIS FAST, the need for such a tool, the costs involved and the way of positioning the tool on the market were be considered.

Applying the diverse criteria method for comparison of the needs for ISIS 2D software (released in 2011) and proposed ISIS FAST software (released in 2011), the estimates from each software were obtained as shown in Table 3. For the purpose of this analysis, in order to obtain objective values when setting the relative weights for each piece of software based on the set of criteria, it was assumed that both pieces of software did not exist. Having these pieces of software taken into account allowed us to obtain the estimates of relative significance of each in comparison with the other.

Based on the weights of the software in Table 3, the following estimates were obtained: 0.684 for ISIS FAST and 0.316 for ISIS 2D. These values were derived using the diverse criteria method described above. First the weights of the criteria were specified. Mostly equal weights were assigned to the criteria, but it was decided that for these two pieces of software the software's competitive appeal was slightly more important than the risk of exceeding the budget. It was also noted that the weight for the criterion 'Costs involved in developing the programming

**Table 3** | Values of weights of the criteria, relative weights and estimates of significance for ISIS 2D and ISIS FAST products

| Criterion | | Compliance of the function with the purpose of the software usage | Technical complexity of the software | Software's contribution to the competitive appeal of the Suite | Costs involved in developing the programming code for this software | Potential problems related to supporting the Suite due to the inclusion of this software | Risks of exceeding the budget when developing programming code for this software | Risks of not meeting the deadline of the next release of the Suite | Estimates of significance of the pieces of software |
|---|---|---|---|---|---|---|---|---|---|
| Weights of the criteria | | 0.14 | 0.14 | 0.2 | 0.14 | 0.14 | 0.1 | 0.14 | |
| Relative weights of software | ISIS FAST | 0.3 | 0.8 | 0.7 | 0.75 | 0.75 | 0.75 | 0.75 | 0.684 |
| | ISIS 2D | 0.7 | 0.2 | 0.3 | 0.25 | 0.25 | 0.25 | 0.25 | 0.316 |
| Coherence rule for relative weights[a] | | ↑ | ↓ | ↑ | ↓ | ↓ | ↓ | ↓ | |

[a]See Table 2 for more details about the rules of coherence.

code' remained unchanged (similar to the most of other criteria). The slightly lower value for the criterion 'Risk of exceeding the budget' meaned that if the budget was to be exceeded and the planned quality of the software obtained, then it would be possible to re-allocate the budget from the other projects, for which estimates according to this method were lower. Then a member of the development and support team provided the values of the weights for each piece of software according to the set of criteria. As the ISIS FAST software was not quite compliant with the purposes of the two-dimensional systems (the most accurate derivation of flooded areas on floodplains) it was given a lower weight according to the compliance criteria.

Due to the simplified approach to the calculation of water levels (ISIS FAST does not use traditional Saint-Venant differential equations), the weights for ISIS FAST was higher according to the criteria 'Technical complexity of the software' and 'Costs involved in developing the programming code for this software' (which due to the opposite meaning of weights for these two criteria means that the code for ISIS FAST is less complicated than the one for ISIS 2D and the development costs were lower). Due to the increased speed of calculation of water levels, the application of a more robust method to ensure water volume conservation (due to the simplified hydraulics used) and acceptable results (which would allow a narrowing down of the area of calculations for the derivation of more precise results), ISIS FAST was considered to have a higher competitive appeal, therefore the weights for ISIS FAST for these criteria were higher. Based on the values of the relative weights of each piece of software according to the pre-defined criteria and the weights of these criteria as specified above, ISIS FAST was assigned a higher relative significance by comparison with ISIS 2D.

The sensitivity tests carried out on the relative weights showed that when the equal weights for both types of software according to the most uncertain criteria (which are 'Possible support problems', 'Risk of exceeding the budget', 'Risk of not meeting the deadline') and perhaps the most influential criterion 'Contribution to competitive appeal' were given, the relative significance of ISIS FAST ($r_{\text{fast}} = 0.549$) was still higher than the relative significance of ISIS 2D ($r_{\text{2D}} = 0.451$).

## Decision-making process for the inclusion of functions in ISIS FAST software

Once the decision to create ISIS FAST was made, there was a need to define its functionality.

The phases of the operation of ISIS FAST software can be broken down into the following three phases: pre-processing, computational engine, post-processing. During the pre-processing phase, such operations as the input of the digital terrain map, reading in DTM data and definition of 'depressions' took place. The computational engine phase followed a set of rules to determine flows over a floodplain. In the post-processing phase such operations as output of flood maps took place. Each phase contained around 10 functions (ISIS Manual 2012). As an example, in this paper the proposed method was applied to the analysis of the functions for inclusion in the post-processing phase of ISIS FAST software. Table 4 below shows the values as per the diverse criteria method.

Once the decision to develop the software has been made, more effort should then be put into developing the software which best matches the purpose of the software. At this stage the purpose of the software was rapid calculation of the water levels on a floodplain and preserving the volume of water in the 2D model. The software should be produced at the lowest cost whilst preserving quality of operation and making sure that the software matches its stated purpose. Therefore the values of the weights of the criteria 'Compliance of the function to the purposes of the software usage' and 'Costs involved in developing the programming code for this software' were the highest. As modifications to the software were planned for the next releases of the Suite and there was uncertainty about the popularity of the software, the problems related to supporting the product could be fixed in the later releases; therefore a lower weight was given to the criterion 'Potential problems related to supporting the product due to the inclusion of this function'. For the same reasons as mentioned above it was possible to re-allocate the budget from other projects with lower estimates of significance, therefore the criterion was also given a slightly lower value. From the table below it is interesting to note that the functions 'Saving output as an ESRI grid', 'Saving output as a Binary Interchange Format grid' and 'Saving output as a shapefile' are more technically complex, but may make a greater contribution to the competitive appeal of ISIS FAST. The overall estimate of the relative significance of these functions was quite low. Due to this, these functions have not yet been included in the releases of the software. However, the functionality of conversion of the files is available in ISIS MAPPER (part of the ISIS Software Suite).

Indeed, the three functions with the highest values of relative significance were included in the first release. The function 'Probable flowpath markup', which received

**Table 4** | Diverse criteria method applied to the decisions about ISIS FAST functions

| Criterion | | Compliance of the function with the purpose of the software usage | Technical complexity of the function | Function's contribution to the competitive appeal of the product | Costs involved in developing the programming code for this function | Potential problems related to supporting the product due to the inclusion of this function | Risk of exceeding the budget when developing programming code for this function | Risk of not meeting the deadlines of the next release of the product | Estimates of significance of the functions |
|---|---|---|---|---|---|---|---|---|---|
| Weights of the criteria | | 0.19 | 0.14 | 0.14 | 0.19 | 0.1 | 0.1 | 0.14 | |
| Relative weights of pieces of software | Generation of a grid of water levels (WL) | 0.3 | 0.175 | 0.1 | 0.175 | 0.2 | 0.27 | 0.25 | 0.21075 |
| | Subtraction of the DTM from the WL and removal of negative values | 0.3 | 0.275 | 0.1 | 0.275 | 0.2 | 0.27 | 0.25 | 0.24375 |
| | Saving output as an ASCII file | 0.1 | 0.2 | 0.1 | 0.2 | 0.19 | 0.2 | 0.2 | 0.166 |
| | Saving output as an ESRI grid | 0.1 | 0.075 | 0.15 | 0.075 | 0.11 | 0.03 | 0.05 | 0.08575 |
| | Saving output as a Binary Interchange Format grid | 0.1 | 0.075 | 0.15 | 0.075 | 0.14 | 0.1 | 0.1 | 0.10275 |
| | Saving output as a shapefile | 0.05 | 0.05 | 0.2 | 0.05 | 0.1 | 0.03 | 0.05 | 0.074 |
| | Probable flowpath markup | 0.05 | 0.15 | 0.2 | 0.15 | 0.06 | 0.1 | 0.1 | 0.117 |
| Rule of coherence[a] | | ↑ | ↓ | ↑ | ↓ | ↓ | ↓ | ↓ | |

[a]See Table 2 for more details.

the fourth highest value of significance, was not included in the first release due to the time pressure for meeting the deadline of the scheduled release, but was subsequently included in the second release of the software approximately 6 months later. This function tries to find the path water may have taken (based on terrain data) to teach the extent of the current flood. The functions which appeared to be problematic according to the support team were improved in the subsequent releases.

## DISCUSSION: DEVELOPMENT AND APPLICATION OF MODELLING SOFTWARE

The above listed examples show that ISIS FAST software has a relatively higher estimate of significance in comparison with ISIS 2D, although the floodplain modelling process still requires ISIS 2D to obtain more precise results and the velocities in the floodplain. The calculation of velocities may be needed where there is a threat to life in the area as the higher velocities pose bigger threat to lives. ISIS FAST and ISIS 2D may produce slightly different results, although as the tests at CH2M HILL showed, ISIS FAST produces the results with a level of precision which can help modellers make a decision if a simulation in a two-dimensional model for the given area is needed. This two-dimensional model can be ISIS 2D or an other 2D floodplain modelling package. The advantage of using ISIS 2D is that the ISIS FAST model can be easily run with ISIS 2D (all the input data are the same, and for ISIS 2D very few additional parameters have to be set, i.e. the time step). The flood maps of a certain area based on the flood calculations using ISIS 2D, shown in Figure 1(a), and ISIS FAST, shown in Figure 1(b), are shown below. These maps show that ISIS FAST can indicate that there will be flooding in the centre of the given area (and so does ISIS 2D). In some areas (especially the lower part of the area) though, ISIS 2D predicts that the flooding extent is likely to be larger than does ISIS FAST. However, taking into account the speed of calculations (several minutes for ISIS FAST and hours for ISIS 2D), it can be said that the main areas of application of ISIS FAST are flood forecasting and national-scale flood mapping (ISIS FAST has been used for both since then). The application of ISIS FAST for flood forecasting sounds promising due to the high speed of calculation and the fact that the results are of acceptable quality for flood forecasting (indicating where the flooding is likely to occur and also of sufficient quality to make decisions for issuing flood warnings). The application of

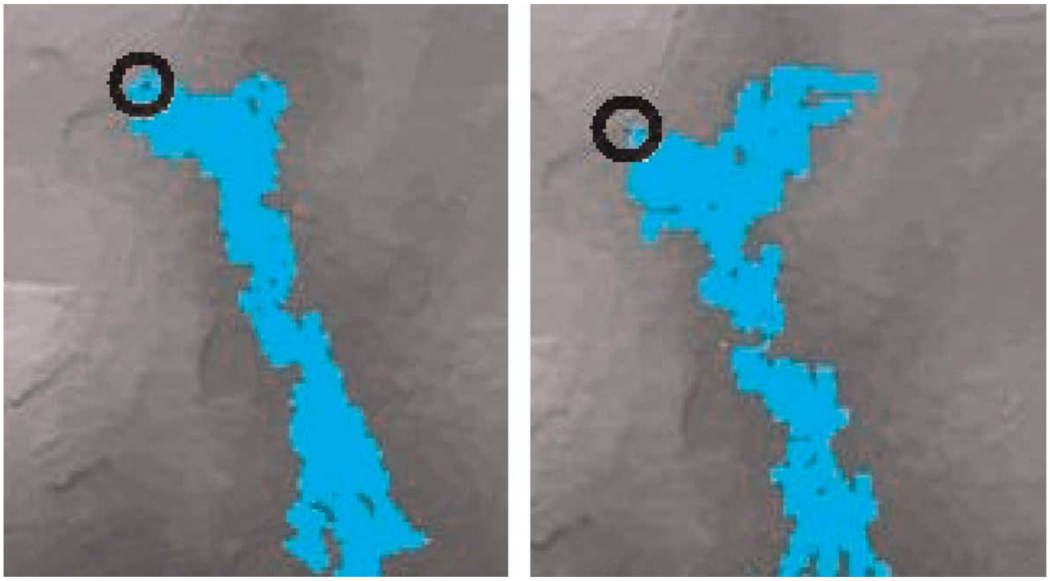

**Figure 1** | Flood extents that result from ISIS 2D (left) and ISIS FAST (right) simulation for the same area (Images courtesy of CH2M HILL). The centre of the in the upper left corner represents critical infrastructure (for example, a hospital or a school).

ISIS FAST for national scale flood mapping seems very reasonable due to again the quick calculation speed of the software and the fact that the results of ISIS FAST allow the modellers to narrow down the calculation area for ISIS 2D, thus saving some time.

ISIS 2D, unlike ISIS FAST, uses Saint-Venant differential equations for the calculations of the water levels on floodplains and in general provides more precise results (and also allows calculation of water velocities). As illustrated earlier, ISIS FAST can be used for rapid screening to help decide where to model in more detail. For example, it is recommended that the ISIS FAST programme be used first to narrow down the calculation area for more detailed analysis using ISIS 2D software. The pictures above show that the calculation area for ISIS 2D can be narrowed down by not including the side areas of the ground grid. For ISIS 2D modelling only the middle two quarters (vertically) of the grid have to be included.

## DISCUSSION: APPLICATION AND FURTHER DEVELOPMENT OF DIVERSE CRITERIA METHOD

The method outlined in this paper can help project leaders make decisions on prioritising the inclusion of functionality into the new software products. The method is applicable to the evaluation of new functions within a single product and making comparisons between two products. The method uses mathematical equations and a set of user-defined weight coefficients of the criteria and relative weight coefficients of the functions to determine the most appropriate functions for inclusion into the software product. The method allows decision-makers to reach the decisions on which functions to include, agreed by all members of the team with the highest level of objectivity. However, the fact that the method requires the experts' judgement may be one of its disadvantages. There are certain measures which can address this disadvantage. In general, the alternative to using experts could be to use qualitative assessment of the data available to support the decision-making (e.g. data available from marketing surveys), but such data for the seven criteria listed above may not always be available at the stage of the requirements analysis. In addition, it should be considered that marketing surveys could be quite expensive. In most cases (definitely in cases with minimum market research data available), the diverse criteria method based on expert judgements is required for verifying the decisions. In order to minimise the subjective element in the judgements, there is a requirement for the experts whose judgement may be different to the others, to substantiate their estimates.

Another challenge for the method is the determination of the weights, which are relative among functions and products. If a new function or a product is to be included then all weights have to be re-evaluated. This challenge can be addressed by asking the experts to rank all the features with the numbers between 1 and 5 (5 being assigned to the function which is the most significant for inclusion according to this criterion; 1 being the least significant), then a calculation, which would average all these numbers and derive a relative value for the weight, could be included in the method. In this case the experts would be able to provide the ranking once for the existing functions and then be asked to provide ranks for new function(s) only.

It should be noted that the diverse criteria method does not take into account the abilities and the skills of the team members to develop a particular function or the capability of selected software tools (programming language, operating system, etc.) to reduce the technical complexity of the function. The method can be improved to address these disadvantages. The multiplicative method of evaluating the relative significance values of the functionality can be implemented in addition to the additive method. This could potentially show a different ranking of the evaluated functions. Software code could be written to simplify the calculations in this method. Alternatively, an existing piece of code, SVIR, produced at Saint Petersburg State Transport University in Russia (www.mcd-svir.ru/refer07.html) can be applied to automating some of the steps of the diverse criteria method. The next step after obtaining the experts' estimates could be the formalisation of this knowledge by means of the methods of ontological modelling (Castañeda *et al.* 2010).

## CONCLUSION

The method outlined in this paper can help software project leaders make decisions about prioritising the inclusion of new functionality into new products or make comparisons between two products. The method uses mathematical equations and a set of user-defined weight coefficients of the criteria and relative weight coefficients of the functions to determine the most appropriate functions for inclusion into the software product. In the paper, seven criteria were identified as important for the selection of the new functionality for a piece of software. The method allows sensitivity tests to be carried out on particular criteria to see which criterion or set of criteria has a stronger impact on the decision made. Such sensitivity tests allow the weights for the most uncertain criteria the be evaluated in a given situation.

The method can be improved by taking into account the abilities and the skills of the team members and the capability of selected software tools and/or programming languages. In addition, the methods of ontological modelling could be used to formalise the knowledge obtained as part of the application of the diverse criteria method described here. Further improvements (ability to use the multiplicative method of evaluation of the weights, removal of the requirement of the sum of the weights to equal 1, etc.) can be implemented in the method. The paper also contains suggestions on how the subjective nature of the fact that the experts give evaluations can be reduced to a minimum in order to provide the most objective evaluations.

Two examples of the application of the method are described in the paper. The first example shows that there is a strong demand for the development of the rapid inundation modelling software ISIS FAST. The second example illustrates the decision-making process regarding the functionality included in the ISIS FAST software. These examples show that the method in its current state is an efficient tool for a team of experts to make decisions regarding which functionality can be included into a new software product or even which product to develop and release first.

## ACKNOWLEDGMENTS

The author wishes to thank David Webb and Yong Wang of CH2M HILL, the editor and the anonymous reviewers for their valuable comments on the draft of the paper.

## REFERENCES

Castañeda, V., Ballejos, L., Caliusco, L. & Galli, R. 2010 The use of ontologies in requirements engineering. *Global Journal of Researches in Engineering* **10** (6), 2–8. (Ver 1.0).

ISIS Manual 2012 Manual for the ISIS Suite of Software Products. V 3.6. CH2M HILL.

Linstone, A. 2002 The Delphi Method. Techniques and Applications. Portland State University. Murray Turoff. New Jersey Institute of Technology.

Volkova, V. N. & Denisov, A. A. 2010 *Methods of the Organization of Complex Expertises*. The Publishing House of Saint Petersburg State Polytechnic University, Saint Petersburg, Russia (in Russian).

Wicks, J., Berry, R., Wilkinson, P., Dunn, B. & Rhodes, F. 2012 New technologies from the UK to support flood risk management. The Proceedings of the Floodplain Management Association Conference, Batemans Bay, NSW, Australia.

First received 22 March 2013; accepted in revised form 8 August 2013. Available online 22 October 2013