

一个基于 GDM 方法的集成软件过程度量模型

郭晓东¹ 孟莉²

(东软集团股份有限公司过程改善中心 沈阳 110179)¹

(东软集团股份有限公司信息规划管理部 沈阳 110179)²

摘要 为满足软件企业不同角色的度量需求,基于 GDM 方法提出了一个面向企业应用的集成软件过程度量模型,模型包含组织结构定义、角色职责定义、GDM 分析结果、业务目标和度量目标评价函数等基础元素,定义了元素间的关联关系。模型支持项目度量元收集范围的确立,维护了组织内数据加工及数据分析流程,可以为角色定制度量分析视图,并辅助确定过程改善活动的目标过程。基于此模型,开发了企业集成软件过程度量系统,在系统支持下,企业度量活动的自动化程度及效率得到了提高。

关键词 软件过程度量模型,软件过程改善,GDM

中图分类号 TP311 文献标识码 A

GDM-based Integrated Software Process Measurement Model

GUO Xiao-dong¹ MENG Li²

(Process Improvement Center, Neusoft Corporation, Shenyang 110179, China)¹

(Information Planning Management Department, Neusoft Corporation, Shenyang 110179, China)²

Abstract To meet the measurement needs of various roles in the software company, based on GDM method, this paper presented an application oriented integrated software process measurement model. The model is established on the organization structure, role responsibility definition, GDM analysis result and evaluation method for the business goal and the measurement goal. It provides the determination of the data collection scope in the project, and the maintenance of the workflow of data processing and data analysis. In addition, the model also supports the construction of the measurement view for organization role, and the identification of the process candidates for process improvement activity. Based on this model, an integrated software process measurement system was developed. As a result, the automation of the measurement activities and measurement efficiency were improved.

Keywords Software process measurement model, Software process improvement, GDM (Goal-Driven Software Measurement)

软件过程度量是重要的软件过程管理活动,是软件工程及过程改善的基础。对于大型软件企业而言,由于其组织结构及决策机制非常复杂,建立合理、有效的过程度量支撑体系尤为重要。目前,软件过程度量相关模型、方法的研究日益成为学术界及企业界的关注重点^[1-10]。在国外,较具代表性的研究成果包括 CMMI^[1], PSM^[2], GQM^[3,4] 和 GDM^[5]。在国内,软件过程度量研究也同样受到了广泛关注^[8-10], 多家研究机构及大学在度量模型建立及度量系统开发方面进行了卓有成效的探索与实践。

本文的研究内容是重点解决企业度量活动面临的实际问题。对于大型软件企业来说,其组织结构非常复杂,各类角色的度量需求存在较大差异,项目级的度量活动往往难以满足企业各级角色的度量需求。为协调企业度量需求(度量消费)与度量实施(度量生产)之间的矛盾,有必要建立一个统一的企业软件过程度量模型,用以规范企业整体软件过程度量活

动。

作者在总结企业度量活动实践经验的基础上,提出了一个企业集成化软件过程度量模型,该模型可以支持企业各类过程度量活动,并在此基础上开发了一个企业集成化软件过程度量系统。本文首先给出集成化软件过程度量模型的模型定义,然后介绍了此模型的多个应用算法,最后简要介绍了基于此模型构建的企业集成化软件过程度量系统的体系结构及功能。

1 企业软件过程度量模型介绍

1.1 企业软件过程度量活动的特点

企业软件过程度量活动有很多特点。首先,企业度量需求具有层次性。由于企业各级组织单元的职能不同,因而各级管理角色的职责及信息需求存在一定差异。企业决策层通常关注:人员的能力是否能够满足业务扩充的需要;客户满意

到稿日期:2008-05-19 本文受 863 国家基金项目(2006AA01Z179)资助。

郭晓东(1968-),男,博士,高级工程师,主要研究方向为软件过程改善、质量管理等, E-mail: guoxiaodong@neusoft.com; 孟莉(1966-),女,硕士,副研究员,主要研究方向为软件过程改善、质量管理等。

度及市场份额是否能够提升;企业盈利能力能否满足投资者的预期。企业实施层通常关注:如何缩短产品上市时间,提升产品的竞争力;如何提升生产率,控制成本。企业操作层(项目)通常关注项目的Q(质量)、C(成本)、D(交付)等指标能否满足合同的要求。对于具体的软件企业,每个层次可能还包含多个子层次,各层次人员的职责和信息需求还会存在一定差异。

其次,企业度量活动是动态的。当企业的外部环境(市场、客户等)和内部环境(人员、技术等)发生变化时,企业各级组织单元的业务目标会做相应的调整,企业整体度量需求也会随之变化。对于一个企业而言,企业各类管理角色是度量数据的消费者,项目组成员及过程保障人员是度量数据的生产者。当企业度量需求发生变化时,项目基础数据的收集活动也要随之调整,以维持企业内部的信息供给/消费平衡。

另外,企业度量活动应面向角色。不仅要保证为各类管理角色提供所需的信息,而且要避免信息的越限使用。实践中,为角色建立度量视图是一个行之有效的方法。度量视图一方面可以限制角色获取度量数据的范围,另一方面可以为角色提供多层次、多角度的分析结果。

除此之外,企业度量活动也是软件过程改善活动的重要组成部分,过程改善目标的确定以及过程改善效果的评估都需要度量活动的支持。

1.2 企业软件过程度量模型结构说明

针对企业软件过程度量活动的特点,作者基于GDM^[3]方法提出了一个面向企业应用的集成软件过程度量模型。GDM方法是美国卡内基·梅隆大学软件工程研究所软件过程度量分析小组(SEMA组)提出的一种面向业务目标的度量分析方法,采用GDM的推荐步骤,可以获得与业务目标对应的度量目标、指示器及度量元。在构造企业软件过程度量体系过程中,会使用GDM分析结果。

图1是企业集成软件过程度量模型。企业是由多级组织单元构成的,每个层级组织单元具有多个业务目标,每个业务目标由特定角色负责。一个业务目标与一组度量目标相对应,每个业务目标和度量目标均有一个评价函数,用于评价业务目标和度量目标的达成情况。度量目标的评价函数是在指示器基础上构造的,指示器对应多个基础度量元。

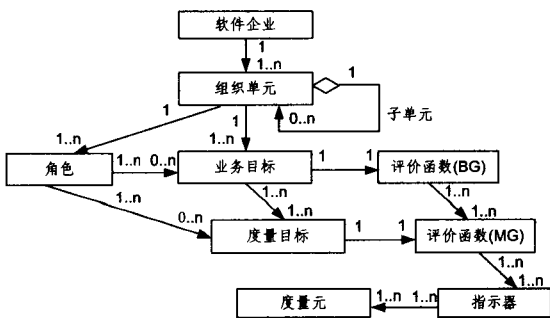


图1 企业集成软件过程度量模型关系图

与企业度量活动几个基本阶段相对应,该模型包含3个组成部分,分别用于度量数据收集、度量数据分析及度量数据使用。其中企业组织结构定义、组织单元角色定义、业务目标和度量目标定义构成了度量数据使用子模型,定义了度量体系的基本框架结构,明确了度量数据的使用者及度量数据的

使用目标;业务目标评价函数、度量目标评价函数及指示器构成了评价子模型,定义了企业度量数据的加工过程,并为角色提供多层次度量分析结果;度量元定义与组织单元定义相关联,定义了度量数据收集子模型,明确了企业范围内需要收集的度量数据以及度量数据的收集范围。

1.3 企业软件过程度量模型定义

本节给出软件过程度量模型的形式化定义,包括业务目标、度量目标、业务目标评价函数、度量目标评价函数、指示器及度量元。同时,为了便于算法描述,定义了一个元组操作。

定义1(取元组元素值操作) 对于元组 $X=(V_1, V_2, \dots, V_n)$, 元素的取值操作定义为 $X.V_i$ 。

定义2(业务目标 Bg) 业务目标(Business goal)是一个四元组, $Bg=(Mg_s, Sp, R_c, E_{Bg})$, Mg_s 是 Bg 对应的度量目标集合, $Mg_s=\{Mg_1, \dots, Mg_n\}$, $Mg_s \subseteq MG$, MG 为组织的度量目标集合; Sp 是 Bg 所属的组织单元; R_c 是关注 Bg 的角色集合, $R_c \subseteq R_{Sp}$, R_{Sp} 是组织单元 Sp 的角色集合; E_{Bg} 是 Bg 的评价函数,用以评价 Bg 是否达成。

业务目标与组织单元相关,组织单元是具有特定职能的业务单元,公司、公司下属业务部门、业务部下属开发中心、项目等都是组织单元。组织单元之间具有隶属关系,公司是最高层级组织单元,项目是企业底层组织单元。定义2中的组织单元 Sp 是一个二元组, $Sp=(Sp_{PF}, Sp_{PS})$, Sp_{PF} 是组织单元 Sp 的上级组织单元。当 $Sp_{PF}=\emptyset$ 时, Sp 为顶层组织单元,即公司级。 Sp_{PS} 是组织单元 Sp 的下级组织单元集合,当 $Sp_{PS}=\emptyset$ 时, Sp 处于组织结构的底层,即项目级。

定义3(度量目标 Mg) 度量目标(Measurement goal)是一个三元组, $Mg=(P, R_c, E_{Mg})$, P 是度量目标关注的过程, $P_r \in P$, P 为组织的过程集合; R_c 是关注度量目标 Mg 的角色集合, $R_c \subseteq R_{Sp}$; E_{Mg} 为度量目标的评价函数,用以评价 Mg 是否达成。

度量目标是业务目标的分解目标。与业务目标相比,度量目标明确了度量活动关注的组织过程及其所要达成的结果,它与具体过程相关。

定义4(业务目标评价函数) $E_{Bg}=(E_s, \varphi(E_s))$, $E_s=\{E_1, \dots, E_n \mid E_i=Mg_i, E_{Mg}, 1 \leq i \leq n, Mg_i \in Bg, Mg_s\}$ 。 E_s 是业务目标 Bg 对应的度量目标的评价函数集合,函数 φ 是基于 E_s 构造的业务目标 Bg 的评价函数。

定义5(度量目标评价函数) $E_{Mg}=(I_s, \theta(I_s))$, I_s 是与度量目标 Mg 相关的指示器集合, $I_s=\{I_1, \dots, I_n\}$, $I_s \subseteq I$, I 为组织的指示器集合, θ 是基于指示器集合 I_s 构造的度量目标 Mg 的评价函数。

业务目标评价函数是在度量目标评价函数基础上构建的。度量目标评价函数是在指示器基础上构建的,评价函数的主要作用是评估业务目标及度量目标的达成情况。通过评价结果可以发现过程活动产生的偏差,并采取应对措施。

定义6(指示器) 指示器(Indicator)是一个二元组, $I_r=(M_s, \gamma(M_s))$, M_s 是与指示器相关的度量元集合, $M_s=\{M_1, \dots, M_n\}$, $M_s \subseteq M$, M 是组织的度量元集合, γ 是基于度量元集合 M_s 的指示器 I_r 的构造函数。

2 企业软件过程度量模型应用

2.1 企业软件过程度量体系的建立

对于组织单元的每个业务目标,采用 GDM 分析方法,可以获得一个由业务目标到度量元的业务目标分析树。在业务目标分析树中,业务目标、度量目标、指示器及度量元之间是一对多的关系。每个组织单元可能包含多个业务目标树,基于组织单元的业务目标分析树,可以构建组织单元度量体系,该体系由业务目标集合、度量目标集合、指示器集合及度量元集合构成,集合元素间是多对多关系,度量体系在组织单元内部维护了一个由度量元到业务目标的数据加工过程。算法 1 用于组织单元度量体系的建立,其中 $B_{G_{S_p}}$ 是业务目标集合, $M_{G_{S_p}}$ 是度量目标集合, I_{S_p} 是指示器集合, M_{S_p} 是度量元集合, $R_{B_{G_{S_p}}}$ 维护业务目标和度量目标的关系, $R_{M_{G_{S_p}}}$ 维护度量目标和指示器的关系, $R_{I_{S_p}}$ 维护指示器和度量元的关系。

算法 1 建立组织单元度量体系

输入:组织单元 S_p 业务目标分析树

输出: S_p 的度量体系 R_{S_p}

算法:

- 1) 将组织单元 S_p 的业务目标纳入 $B_{G_{S_p}}$;
for $B_g \in B_{G_{S_p}}$,
{ $M_{G_{S_p}} = M_{G_{S_p}} \cup B_g, M_{G_s}$;
根据 B_g, M_{G_s} , 维护 $R_{B_{G_{S_p}}}: B_{G_{S_p}} \rightarrow M_{G_{S_p}}$;
}
- 2) for $M_g \in M_{G_{S_p}}$,
{ $I_{S_p} = I_{S_p} \cup M_g, E_{M_g}, I_s$;
根据 M_g, E_{M_g}, I_s , 维护 $R_{M_{G_{S_p}}}: M_{G_{S_p}} \rightarrow I_{S_p}$;
}
- 3) for $I_r \in I_{S_p}$,
{ $M_{S_p} = M_{S_p} \cup I_r, M_s$;
根据 I_r, M_s , 维护 $R_{I_{S_p}}: I_{S_p} \rightarrow M_{S_p}$;
}
- 4) 构建 S_p 的度量体系,
 $R_{S_p} = (B_{G_{S_p}}, M_{G_{S_p}}, I_{S_p}, M_{S_p}, R_{B_{G_{S_p}}}, R_{M_{G_{S_p}}}, R_{I_{S_p}})$

算法结束。

2.2 项目度量元收集范围的确立及维护

企业度量活动的一个关键环节是项目度量元的确立与维护。项目度量活动不仅要满足项目管理的信息需求,也要满足组织各级管理角色的信息需求,并且应随组织业务目标的变化随时调整。

算法 2 在项目启动阶段为项目确定度量元收集范围。该算法首先基于项目业务目标分析树,构造项目度量体系,获得项目度量元集合。然后依次向上寻找各级组织单元的度量元集合,并在此基础上构造项目度量元收集范围。当搜索到企业顶层组织单元时,算法结束。图 2 是确定项目度量元收集范围的示例,示例中的企业有 4 个组织层级,项目度量元收集范围包含项目及其上级组织单元的度量元集合(虚线范围内的度量元集合)。

算法 2 建立项目度量元收集范围

输入:项目 P 的业务目标分析树, P 的上级组织单元度量体系

输出:MP(项目度量元收集范围)

算法:

- 1) 基于 P 的业务目标分析树,调用算法 1,构造 P 的度量体系 R_P ;
将 R_P 度量元集合纳入 MP, $MP = R_P, M_{S_p}$;
 $S_p = P$;
- 2) 取组织单元 S_p 的父组织单元, $S_p = S_p, S_{PF}$;
获取 S_p 的度量体系 R_{S_p} ;

$M_P = M_P \cup R_{S_p}, M_{S_p}$;

if ($S_p, S_{PF} \neq \emptyset$)

返回(2)继续执行;

算法结束。

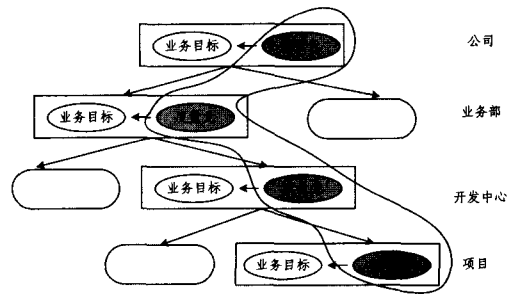


图 2 建立项目度量元收集范围

当某个层级组织单元业务目标发生变化(包括增、删、改)时,可能导致所辖项目的度量元收集范围发生变更。算法 3 以某组织单元新增一个业务目标为例,说明业务目标变化如何影响项目度量元收集范围。该算法首先确定新增业务目标对应的度量元集合,然后确定该组织单元所辖的项目范围,并基于新增业务目标的度量元集合,调整受影响项目的度量元收集范围,最后维护组织单元的度量体系。以图 2 为例,当企业某个层级业务单元的业务目标发生变化时,该业务单元所辖的所有项目的度量元收集范围也会随之改变。

算法 3 维护项目度量元收集范围和度量体系

输入:新增业务目标 B_g 的分析树, S_p 的度量体系 R_{S_p}

输出: S_p 所辖项目的度量元收集范围, S_p 的度量体系 R_{S_p}

算法:

- 1) 调用算法 1,构造与新增 B_g 对应的指标体系 $R_{B_g}, R_{B_g} = (B_g, M_{G_{B_g}}, I_{B_g}, M_{B_g}, R_{B_{G_{B_g}}}, R_{M_{G_{B_g}}}, R_{I_{B_g}})$
- 2) 调用算法 4,确定 S_p 所辖的所有项目 S_{P_j} ,
 $S_{P_j} = \text{GetAffectedProjects}(S_p)$;
for $P_j \in S_{P_j}$,
调整 P_j 的度量范围 $M_{P_j}, M_{P_j} = M_{P_j} \cup R_{B_g}, M_{B_g}$;
- 3) 基于 R_{B_g} ,重构 S_p 度量体系 R_{S_p} 。

算法结束。

算法 4 被算法 3 调用,它为组织单元确定所辖的项目集合。算法 4 由组织单元开始,向下递归查询所辖项目, U_i 是组织单元 S_p 的下级组织单元,当 U_i 的下级组织单元为空时, U_i 是企业的底层组织单元,即项目。

算法 4 确定组织单元 S_p 所辖项目

名称: $\text{GetAffectedProjects}(S_p)$

输入:组织单元 S_p

输出: S_p 所辖的项目集合 S_{P_j}

算法:

- for $U_i \in S_p, S_{PS}$,
- if ($U_i, S_{PS} = \emptyset$)
- $S_{P_j} = S_{P_j} \cup U_i$;
- else
- $S_{P_j} = S_{P_j} \cup \text{GetAffectedProjects}(U_i)$;

算法结束。

2.3 建立角色度量视图

组织度量活动的一个重要环节是度量结果的使用,信息

查询方式是制约度量结果有效使用的一个重要因素。实际工作中,通常是数据使用者在不同工具和报告中搜寻信息,而不是通过一种有效渠道获得信息。集成化软件过程度量模型可以为解决这一问题提供帮助,模型中业务目标及度量目标是与角色相关的,可以据此为每个角色定制一个度量视图。视图中包含角色关注的业务目标、度量目标及指示器分析结果,这样就可以在工具的支持下将角色关注的信息“推”到使用者面前。

算法 5 为组织角色建立度量视图,角色可能关注 3 类度量信息:业务目标、度量目标、指示器。当角色关注某个业务目标时,要将该业务目标相关的所有度量目标评价函数及指示器都纳入度量视图。当角色关注的只是业务目标下的某个度量目标时,仅将该度量目标评价函数及相关指示器纳入度量视图。算法 5 首先确定角色所关注的业务目标评价函数,然后维护其所关注的度量目标集合,最后根据度量目标集合确定度量目标评价函数及指示器。算法中的 M_{GR} 是角色关注的度量目标集合。

算法 5 为组织单元角色建立度量视图

输入:组织单元的角色 R ,组织单元度量体系 R_{Sp}

输出: R 的度量视图 V_R

算法:

```

1)for  $Bg \in R_{Sp}, Bg_{Sp}$ ,
   if( $R \in Bg, R_c$ )
       {  $V_R = V_R \cup \{Bg, E_{Bg}\}$ ;
          $M_{GR} = M_{GR} \cup Bg, M_{GS}$ ;
       }
2) $M_{GT} = R_{Sp}, M_{GS} - M_{GR}$ ;
   for  $Mg \in M_{GT}$ ,
       if( $R \in Mg, R_c$ )
            $M_{GR} = M_{GR} \cup \{Mg\}$ ;
3)for  $Mg \in M_{GR}$ 
   {  $V_R = V_R \cup \{Mg, E_{Mg}\}$ ;
      $V_R = V_R \cup \{Mg, E_{Mg}, I_s\}$ ;
   }

```

算法结束。

2.4 确定改善活动目标过程

过程改善的目的是提升组织单元的过程能力,以便达成组织单元的业务目标。在集成软件过程度量模型中,分别为业务目标和度量目标定义了评价函数。当评价结果异常时,预示着组织设定的目标没有达成,或过程性能产生了偏差,相关异常的过程可以作为过程改善活动的关注重点。算法 6 根据业务目标和度量目标的评价结果,为组织单元确定候选改善过程。 M_{GB} 是异常业务目标对应的度量目标集合, M_{GM} 是异常的度量目标集合,两者的交集是异常业务目标对应的异常度量目标集合(异常业务目标的某些度量目标可能是正常的),此类度量目标相关过程的改善优先级较高。

算法 6 确定组织单元 S_p 的候选改善过程

输入:组织单元度量体系 R_{Sp}

输出:候选改善过程 P_{Sp}

算法:

```

1)for  $Bg \in R_{Sp}, Bg_{Sp}$ ,
   if( $Bg, E_{Bg}$  异常)

```

$M_{GB} = M_{GB} \cup Bg, M_{GS}$;

2)for $Mg \in R_{Sp}, M_{GS}$,

if(Mg, E_{Mg} 异常)

$M_{GM} = M_{GM} \cup \{Mg\}$;

3)for $Mg \in M_{GM}$,

if($Mg \in (M_{GB} \cap M_{GM})$)

$P_{Sp} = P_{Sp} \cup \{(H, Mg, P_r)\}$;

else

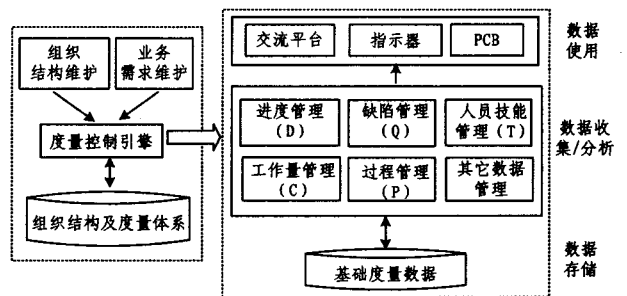
$P_{Sp} = P_{Sp} \cup \{(L, Mg, P_r)\}$;

算法结束。

3 企业集成软件过程度量系统的建立与应用

东软集团是国内领先的软件企业,其业务涵盖了软件开发、解决方案、外包等多个领域,业务规模及人员规模庞大,给软件过程管理带来了巨大挑战。为适应业务拓展及内部管理的需要,东软集团建立了多个业务群组,每个业务群组包含多个相对独立的业务单元,每个业务单元根据业务及客户需求构建了多级管理机制。如此复杂的组织结构,对软件过程信息管理提出了很高要求,既要保证各级管理者及时了解软件过程信息,还要努力减少度量活动产生的附加工作量。

东软集团重视软件过程管理的规范化。在实施软件过程管理过程中,针对质量管理、工作量管理等业务需求,陆续开发了多个供项目使用的管理工具。为适应业务规模的不断扩充,提高企业整体的实施效率,公司针对度量过程进行了全面梳理。基于企业集成化软件过程度量模型,对原有工具系统进行了重新整合,构造了一个企业集成软件过程度量系统,体系结构如图 3 所示。整个系统从架构上分为度量体系维护及度量活动实施两个部分,度量体系维护的主要工作是构造适应组织结构、角色定义及业务需求变化的企业软件过程度量体系,用以控制企业度量活动的实施。度量活动实施部分包括度量数据存储、度量数据收集/分析及度量数据使用 3 个层次,每个层次包含多个功能系统,支持企业集成化软件过程度量活动。



度量体系维护系统包括 2 个接口功能:组织结构维护和业务需求维护,它们支持组织结构及业务需求的定义及维护。基于企业组织结构和业务需求,度量控制引擎构造企业软件过程度量体系,并据此控制整个企业的度量活动。

度量活动实施系统包括数据使用、数据收集与分析和数据存储 3 个层次。在数据收集与分析层包括几个常用工具,分别用于质量(Q)、工作量(C)、进度(D)、过程保障(P)、人

(下转第 162 页)

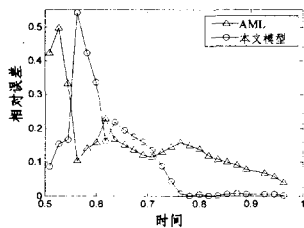


图8 Windows 2003 漏洞预测的相对误差

结束语 软件的安全漏洞本质上是一种缺陷,因此软件漏洞检测过程与软件故障检测过程有一定程度的相似之处。另一方面,由于软件故障检测和软件漏洞检测的目的、参与人员和发生的时间不尽相同,描述这两个过程的模型又会有一些不同。本文深入分析了影响软件安全漏洞检测的因素。与现有的安全漏洞检测模型不同的是,本文认为软件的市场占有率也是影响软件安全漏洞检测速度的一个重要因素,并在此基础上建立了基于市场占有率的漏洞检测模型。通过分析模型的特性,我们得到了两个重要结论:一是在软件发布之前已经暴露了一些安全漏洞;二是并不是所有的安全漏洞都会被检测出来。这两个结论在实际的数据中得到了验证。我们用本文提出的模型分析了3种流行操作系统的漏洞检测数据集,得出了这3种操作系统的漏洞检测还处于快速上升阶段的结论。与AML相比,大多数情况下本文提出的模型有更好的拟合能力与预测能力。

参考文献

[1] Zhang W F, Tao Xin Li, Jose F. An Analysis of Microarchitec-

ture Vulnerability to Soft Errors on Simultaneous Multithreaded Architectures//IEEE International Symposium on Performance Analysis of Systems & Software, 2007 (ISPASS 2007). 2007

[2] Salas P A P, Krishnan P, Ross K J. Model-based Security Vulnerability Testing//18th Australian Software Engineering Conference 2007 (ASWEC 2007). 2007

[3] Savola R K, Kaarina. Practical Security Testing of Telecommunications Software—A Case Study//The Third Advanced International Conference on Telecommunications 2007 (AICT 2007). 2007

[4] Schultz J E, Brown D, Longstaff T. Responding to computer security incidents. Lawrence Livermore National Laboratory, 1990. Ftp://ftp.cert.dfn.de/pub/docs/csir/ihg.ps.gz

[5] Goel A L, Okumoto K. Time-dependent error-detection rate model for software and other performance measures. IEEE Transaction on Reliability, 1979, 28: 206-211

[6] 刘宏伟, 杨孝宗, 曲峰, 等. 一个基于响铃形故障检测率函数的软件可靠性增长模型[J]. 计算机学报, 2005, 28(5): 908-913

[7] Alhazmi O H, Malaiya Y K. Application of Vulnerability Discovery Models to Major Operating Systems. IEEE Trans. Reliability, 2008; 14-22

[8] Alhazmi O H, Malaiya Y K, Ray I. Measuring, Analyzing and Predicting Security Vulnerabilities in Software Systems. Computers and Security Journal, 2007; 26(3): 219-228

[9] Alhazmi O H, Malaiya Y K. Measuring and Enhancing Prediction Capabilities of Vulnerabilities Discovery Models for Apache and IIS HTTP Servers//Symp. Software Reliability Eng. 2006; 343-352

[10] National Vulnerability Database. <http://nvd.nist.gov/>

(上接第158页)

员/技能(T)的管理,另外还为收集其它数据构建了接口,以满足其它软件过程度量需求。数据使用层为各级角色定制了度量视图,提供与角色度量需求相适应的数据分析结果。另外,为支持度量数据有效使用,数据使用层还提供了各组织单元的PCB和度量活动交流平台。图4为企业集成软件过程度量系统启动画面。

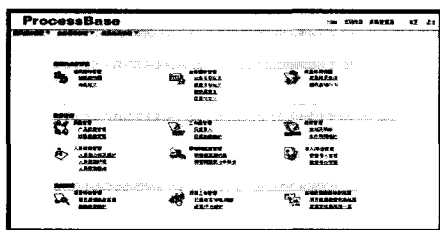


图4 企业集成软件过程度量系统启动画面

结束语 针对软件企业过程度量活动的特点,作者提出了一个企业集成软件过程度量模型。该模型满足企业度量活动各阶段要求,使企业的度量生产及度量消费协调一致,同时该模型可以支持过程改善活动的顺利开展。

基于企业集成软件过程度量模型,我们开发了一个集成化软件过程度量系统,该系统支持企业过程度量数据的收集、分析及使用,提供项目软件过程动态监控及业务单元经营状态分析,满足企业不同层级角色的过程度量需求。在系统支持下,企业度量活动的目的性及自动化程度均有很大提升。

下一步工作重点是研讨企业软件过程度量信息与企业其

它业务信息的融合方法,拓展其应用范围,加强企业间各类信息的互通,减少数据的冗余及不一致性,简化各类角色获取信息的通路,提升软件企业信息化整体水平。

参考文献

[1] CMMI Product Team. Capability Maturity Model Integration (CMMI) for development Version 1. 2. SEI CMU/SEI-2006-008(ESC-TR-2006-008). 2006. 8

[2] McGarry J, Card D, Jones C, et al. Practical Software Measurement: Objective Information for Decision Makers[M]. Addison-wesley, 2002

[3] Solingen R, Berghout E. The Goal/Question/Metric Method: a Practical Guide for Quality Improvement of Software Development[M]. The McGRAW-HILL Companies, 1999

[4] Basili V R, Caldiera G, Rombach H D. Goal Question Metric Paradigm// Marciniak J J, ed. Encyclopedia of Software Engineering. New York: John Wiley & Sons, Inc., 1994: 528-532

[5] Park R E, Goethert W B, Florac W A. Goal-driven Software Measurement—A Guidebook. SEI CMU/SEI-96-HB-002. 1996. 8

[6] Lawler J, Kitchenham B. Measurement modeling technology. IEEE Software, 2003, 20(3): 42-48

[7] Junlin Huang, Far B H. Intelligent Software Measurement System for Automating the Goal-Question-Metrics Process// Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06). 2006

[8] 钱红兵, 朱丽娟, 曹惠民. 基于CMM的软件过程度量系统的研究与设计[J]. 计算机应用研究, 2004, 21(6): 49-52

[9] 任发科, 周伯生, 吴超英. 软件度量过程的研究与实施[J]. 北京航空航天大学学报, 2003, 29(10): 931-934

[10] 王青, 李明树, 刘霞. 一种支持软件过程控制和改进的主动度量模型[J]. 软件学报, 2005, 16(3): 407-418