Software as a Service: Configuration and Customization Perspectives

软件即服务：配置与定制观点

Wei Sun, Xin Zhang, Chang Jie Guo, Pei Sun, Hui Su IBM China Research Lab, Beijing 100094, P.R. China {weisun, zxin, guocj, sunpei, [suhui}@cn.ibm.com](mailto:suhui%7d@cn.ibm.com)

Abstract

Software as a Serivce(SaaS) provides software application vendors a Web based delivery model to serve big amount of clients with multi-tenancy based infrastructure and application sharing architecture so as to get great benefit from the economy of scale. Though SaaS application is usually developed with highly standardized software functionalities to serve as many clients as possible, many clients still ask for function variants according to their unique business needs through easy configuration and customization. Due to the subscription based model, SaaS vendors need take a well designed strategy to enable self serve configuration and customization by their customers without changing the SaaS application source code for any individual customer. In this paper, we will explore the configuration and customization issues and challenges to SaaS vendors, clarify the difference between configuration and customization. A competency model and a methodology framework have been developed to help SaaS vendors to plan and evaluate their capabilities and strategies for service configuration and customization.

摘要

软件即服务（saas）为软件应用程序供应商提供基于网络的交付模式，以便为基于多租户的基础架构和应用程序共享架构的大量客户提供服务，从而从规模经济中获得巨大收益。尽管saas应用程序通常采用高度标准化的软件功能开发，以尽可能多地服务于客户，但许多客户仍然通过简单的配置和定制，根据其独特的业务需求获取特定功能。由于基于订阅的模型，Saas供应商需要采取精心设计的策略来实现客户的自助服务配置和定制，而无需为了任何单个客户更改Saas应用程序源代码。在本文中，我们将探讨Saas供应商的配置和定制问题以及挑战，阐明配置和定制之间的差异。因此我们开发了有力模型和方法论框架，以帮助Saas供应商计划和评估其服务配置和定制的功能和策略。

1. Introduction

Software as a Service(SaaS) is gaining momentum with the significant increased number of vendors moving into this space and the recent success of a bunch of leading players on the market[1][2]. Designed to leverage the benefits brought by economy of scale, SaaS is about delivering software functionalities to a big group of clients over Web with one single instance of software application running on top of a multi-tenancy platform[3]. Clients usually don’t need to purchase the software license and install the software package in their local computing environment. They use the credentials issued by the SaaS vendor to log onto and consume the SaaS service over Web through an Internet browser at any time and from any where with Internet connections. As well known, in the enterprise software area, software is designed to support a business’ operations in terms of business data management, process automation and optimization, and governance. SaaS is not an exceptional case though it does provide major differentiating advantages around much lower Total Cost of Ownership(TCO) and stronger mobility allowed for the users through the Web based delivery model[4]. However, every client is unique which leads into the requirements’ variance to the software. The fundamental causes of the requirements variance among clients include: industry focus differences; customer behavior differences; product offering differences; regulation differences; culture differences and operation strategy differences. Therefore most of enterprise software applications need to be tailored more or less so as to effectively serve a specific client. The widely used approaches of tailoring software are configuration and customization. Software configuration and customization is not a brand new problem. There are many academic research and industrial best practices available already, for example: Software Configuration Management(SCM) theory was developed by Roger Pressman through software engineering research[5]; SAP software applications have strong configuration and customization capabilities through Graphic User Interface(GUI) based tool and script based programming tool(ABAP)[6]. The leading SaaS vendors have developed profound configuration an customization capabilities as well, for example, Salesfoce.com provides Apex to facilitate the extensive application configuration and customization on the Web based on the multi-tenancy architecture.[7] As well understand, the delivery model of SaaS does not allow the SaaS vendor to develop and maintain a version of application code for each individual customer, the configuration and customization capabilities of SaaS service play an extremely important role for its success. However there is not a thorough study of the configuration and customization issues of SaaS from both business and technology perspectives to guide a SaaS vendor to plan and execute strategies around configuration and customization. In the section 2 of the paper, we explore the difference between configuration and customization, as well as the challenges in the SaaS multi-tenancy environment; In section 3, we introduce the configuration and customization competency model and also the analysis of the aspects which are usually demanded for configuration and customization; In section 4, we present a methodology framework to guide SaaS vendor to plan and execute configuration and customization strategies.

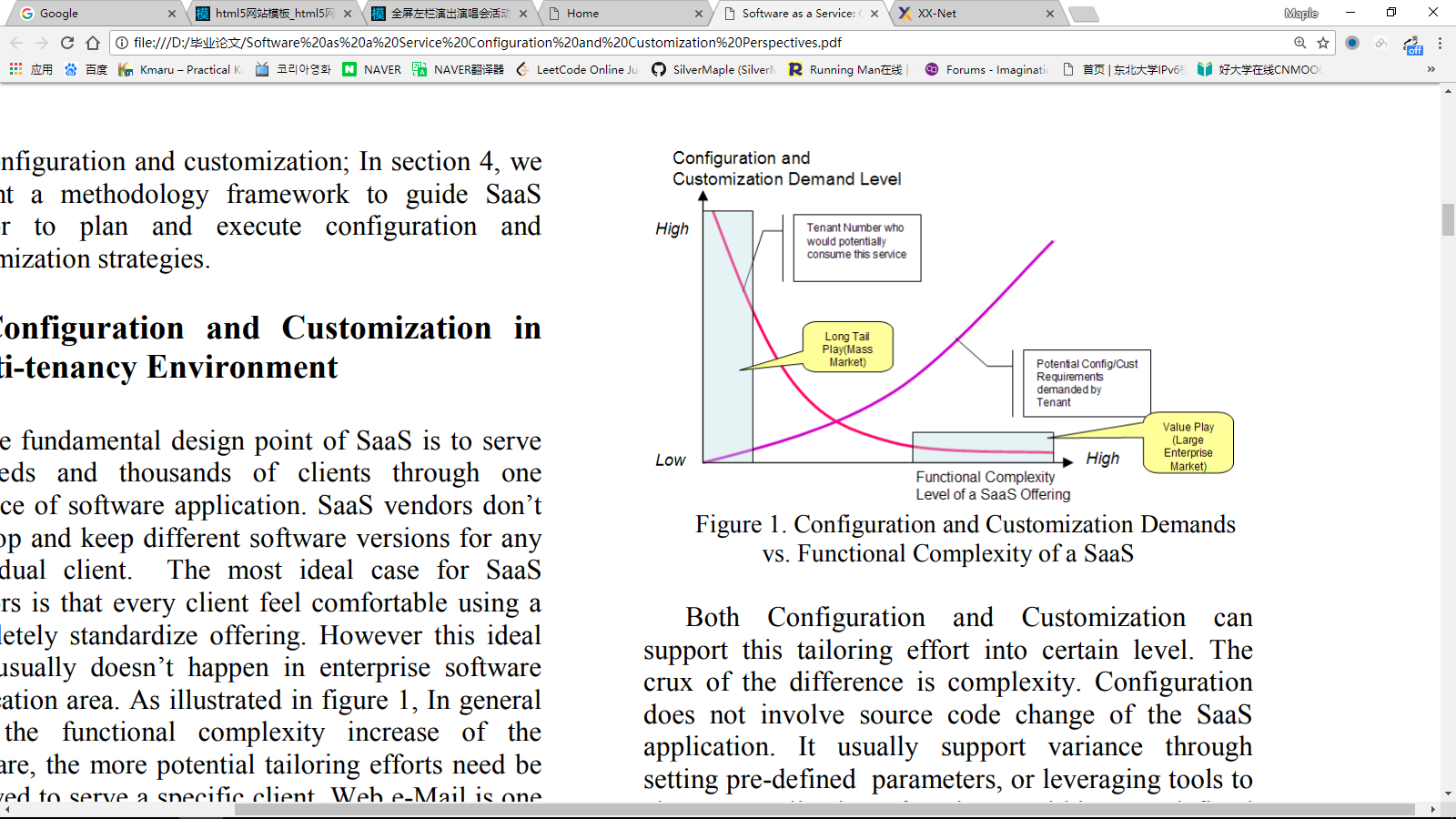
1. 简介

软件作为一项服务（saas）的势头越来越强劲，随着越来越多的供应商进入这个领域以及最近在市场上一批领先企业的成功[1] [2]。旨在充分利用规模经济带来的优势，saas就是通过在多租户平台上运行的单一软件应用程序实例将软件功能交付给一大群客户。客户通常不需要购买软件许可证并将软件包安装在本地计算环境中。他们使用saas供应商发布的凭证，通过互联网浏览器随时登录和使用saas服务，并从任何地方使用互联网连接。众所周知，在企业软件领域，软件旨在支持业务数据管理，流程自动化和优化以及治理方面的业务运营。尽管Saas在总体拥有成本（tco）和用户通过基于Web的交付模式方面提供了更强大的移动性，但它确实提供了主要的差异化优势[4]。然而，每个客户都是独一无二的，这导致了软件需求的变化。客户需求差异的根本原因包括：行业重点差异;顾客行为差异;产品提供差异;监管差异;文化差异和经营策略差异。因此大多数企业软件应用程序需要或多或少地进行定制，以便有效地为特定客户提供服务。裁剪软件的广泛使用的方法是配置和定制。软件配置和定制不是一个全新的问题。已有许多学术研究和行业最佳实践，例如：roger pressman通过软件工程研究[5]开发了软件配置管理（scm）理论;sap软件应用程序通过基于图形用户界面（gui）的工具和基于脚本的编程工具（abap）[6]具有强大的配置和定制功能。领先的SaaS供应商也开发了深度配置和定制功能，例如，salesfoce.com提供了一个基于多租户体系结构的Web上广泛的应用程序配置和定制[7]。我们也明白，saas的交付模式不允许saas供应商针对每个客户开发和维护一个应用程序代码版本，saas服务的配置和定制功能对其成功起着非常重要的作用。然而，从业务和技术角度来看，这并没有深入研究saas的配置和定制问题，以指导saas供应商围绕配置和定制计划和执行策略。在本文的第2部分中，我们探讨了配置和定制之间的区别，以及Saas多租户环境中的挑战;在第3节中，我们介绍了配置和定制能力模型，以及通常要求的配置和定制方面的分析;在第4节中，我们提供了一个方法框架来指导Saas供应商计划和执行配置和定制策略。

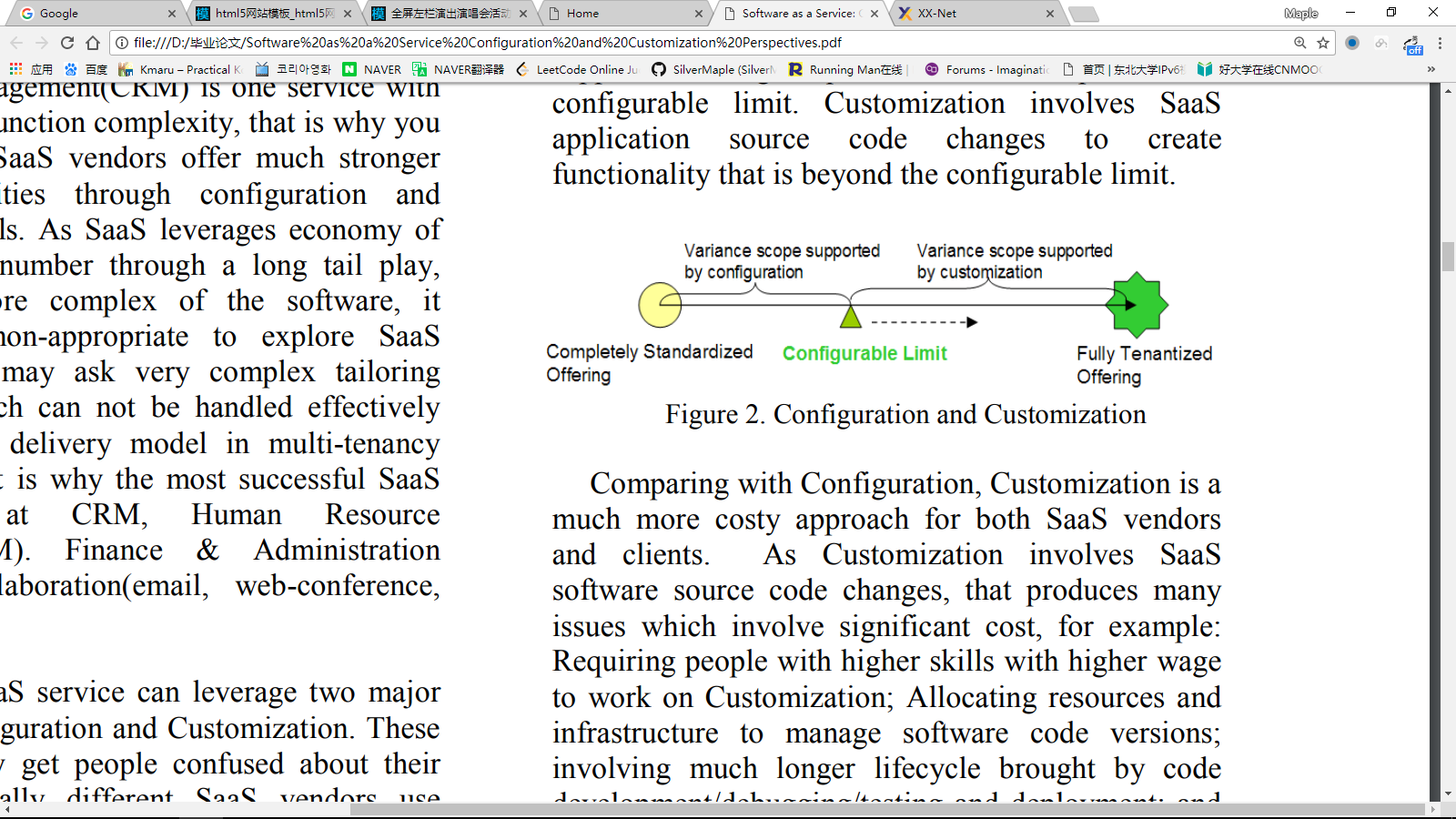
1. Configuration and Customization in Multi-tenancy Environment

The fundamental design point of SaaS is to serve hundreds and thousands of clients through one instance of software application. SaaS vendors don’t develop and keep different software versions for any individual client. The most ideal case for SaaS vendors is that every client feel comfortable using a completely standardize offering. However this ideal case usually doesn’t happen in enterprise software application area. As illustrated in figure 1, In general with the functional complexity increase of the software, the more potential tailoring efforts need be involved to serve a specific client. Web e-Mail is one SaaS service with relatively simple functions, that is why clients usually just need to tailor the service with parameters based setting, e.g. e-mail box storage size; account number. Industry generic Customer Relationship Management(CRM) is one service with medium level of function complexity, that is why you see many CRM SaaS vendors offer much stronger tailoring capabilities through configuration and customization tools. As SaaS leverages economy of scale of clients’ number through a long tail play, therefore the more complex of the software, it becomes more non-appropriate to explore SaaS model as client may ask very complex tailoring requirements which can not be handled effectively with Web based delivery model in multi-tenancy environment. That is why the most successful SaaS services stay at CRM, Human Resource Management(HRM). Finance & Administration (F&A) and Collaboration(email, web-conference, etc) spaces.[8]

Tailoring a SaaS service can leverage two major approaches: Configuration and Customization. These two terms usually get people confused about their differences. Actually different SaaS vendors use different terms in different contexts. We try to clarify the difference between them. As depicted in figure 2, In order to make a standardized SaaS offering to serve a specific client, we need to tailor it into a tenantized offering by satisfying this client’s unique requirements.



Both Configuration and Customization can support this tailoring effort into certain level. The crux of the difference is complexity. Configuration does not involve source code change of the SaaS application. It usually support variance through setting pre-defined parameters, or leveraging tools to change application functions within pre-defined scope, e.g. adding data fields, changing field names, modifying drop-down lists, adding buttons, and changing business rules, etc. Configuration can support tailoring requirements within pre-defined configurable limit. Customization involves SaaS application source code changes to create functionality that is beyond the configurable limit.



Comparing with Configuration, Customization is a much more costy approach for both SaaS vendors and clients. As Customization involves SaaS software source code changes, that produces many issues which involve significant cost, for example: Requiring people with higher skills with higher wage to work on Customization; Allocating resources and infrastructure to manage software code versions; involving much longer lifecycle brought by code development/debugging/testing and deployment; and losing business opportunity from those clients who can not accept the Customization complexity and cost[9]. The Customization is becoming much more complex in SaaS context, as SasS vendors need to maintain every piece of Customization code tenant by tenant. Upgrading the SaaS application should not lead into losing of any single tenant’s customization code. Therefore wherever possible, SaaS should avoid Customization by using Configuration to meet clients’ tailoring requirements and enlarge configurable limit as far as possible toward client’s unique requirements.

1. 多租户环境中配置和定制

saas的基本设计要点是通过一个软件应用程序实例为成千上万的客户提供服务。Saas供应商不为任何个人客户开发和保留不同的软件版本。对于Saas供应商来说，最理想的情况是每个客户都可以使用完全标准化的产品。然而这种理想情况通常不会发生在企业软件应用领域。如图1所示，通常随着软件功能复杂性的增加，需要更多潜在的裁剪工作来为特定客户提供服务。网络电子邮件是具有相对简单功能的一种saas服务，这就是为什么客户通常只需要用基于参数的设置来定制服务的原因，例如，电子邮箱存储大小;帐号。行业通用客户关系管理（crm）是一种具有中等功能复杂性的服务，这就是为什么您会发现许多crm saas供应商通过配置和定制工具提供更强大的定制功能。由于随着saas通过长尾巴游戏利用客户数量的规模经济，，所以软件越复杂，探索saas模型变得越不合适，因为客户可能会要求非常复杂的裁剪需求，这些需求无法通过web在多租户环境中基于网络的交付模式有效处理。这就是为什么最成功的Saas服务仍然停留在人力资源管理（HRM）。财务与行政（f＆a）和协作（电子邮件，网络会议等）空间。[8]

定制saas服务可以利用两种主要方法：配置和定制。这两个词通常让人们对他们的分歧感到困惑。实际上不同的Saas供应商在不同的环境下使用不同的术语。我们试图澄清它们之间的区别。如图2所示，为了制定标准化的Saas产品服务于特定的客户，我们需要通过满足客户的独特要求，将其定制为一种零售产品。

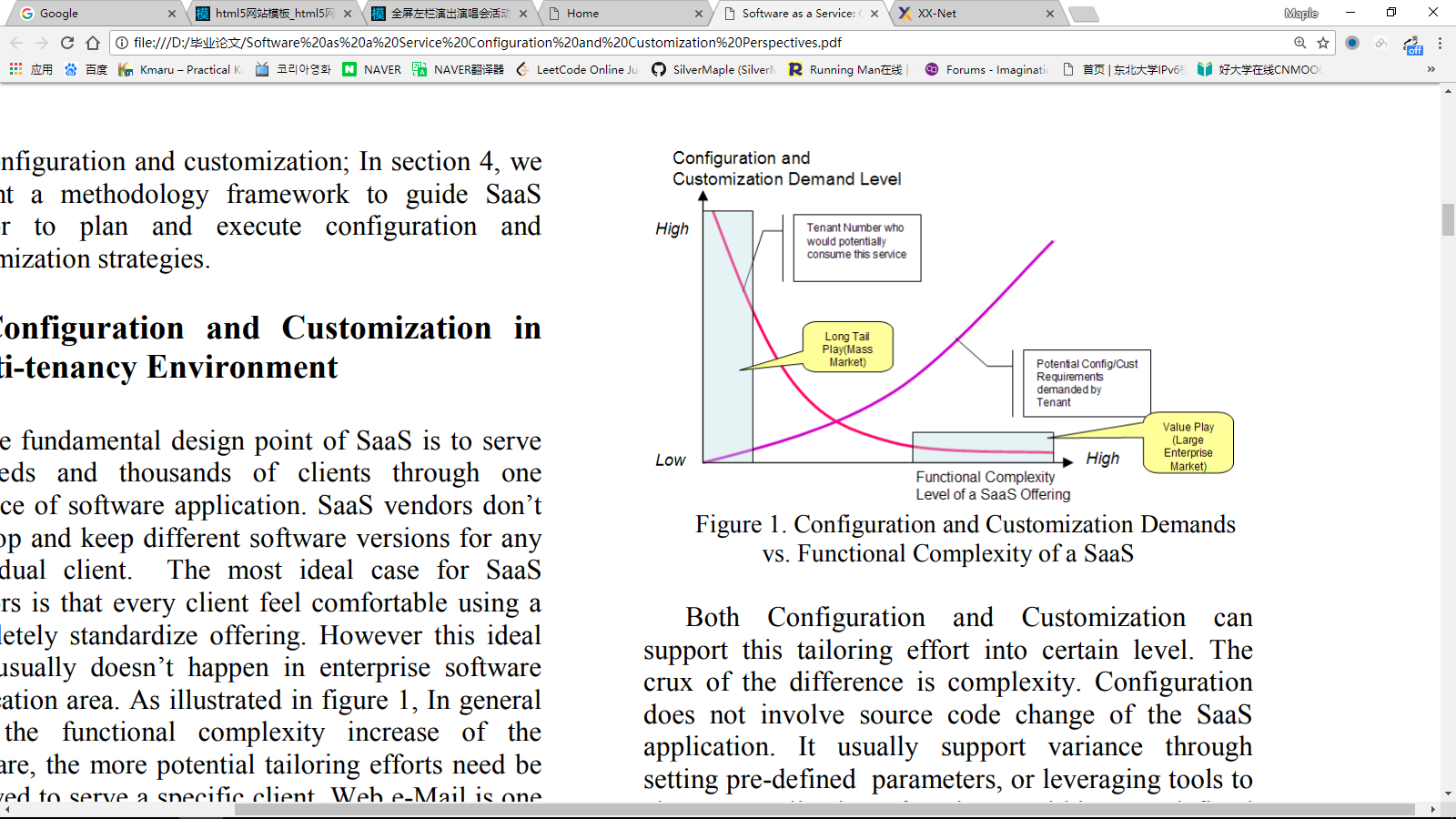


图1.配置和定制需求与saas配置和定制的功能复杂性

配置和定制都可以支持这种定制工作到一定水平。差异的关键在于复杂性。配置不涉及saas应用程序的源代码更改。它通常通过设置预定义的参数来支持差异，或者利用工具来改变预定范围内的应用程序功能，例如，添加数据字段，更改字段名称，修改下拉列表，添加按钮以及更改业务规则等。配置可以支持在预定义的可配置限制内定制需求。定制涉及saas应用程序源代码更改以创建超出可配置限制的功能。

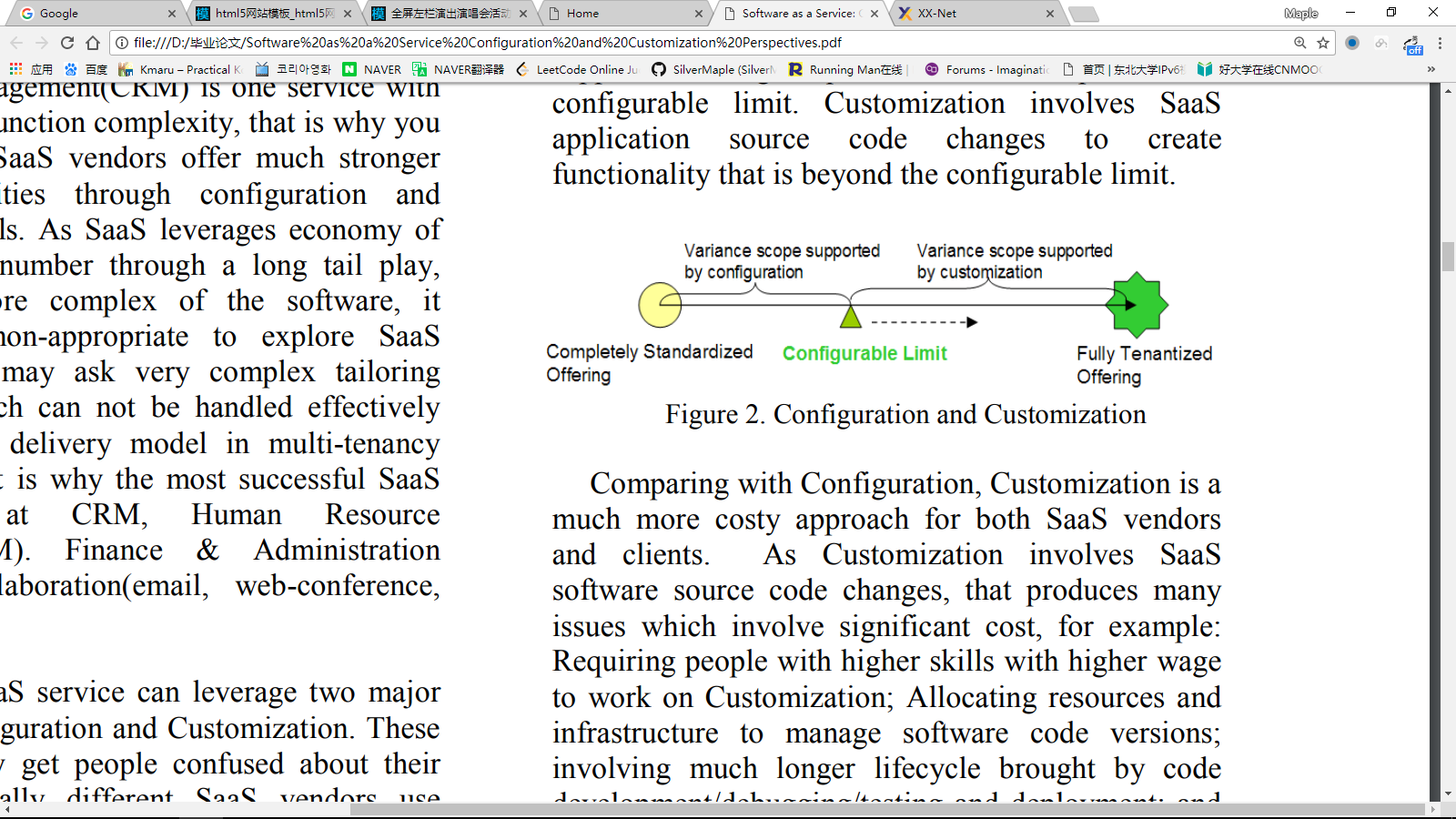
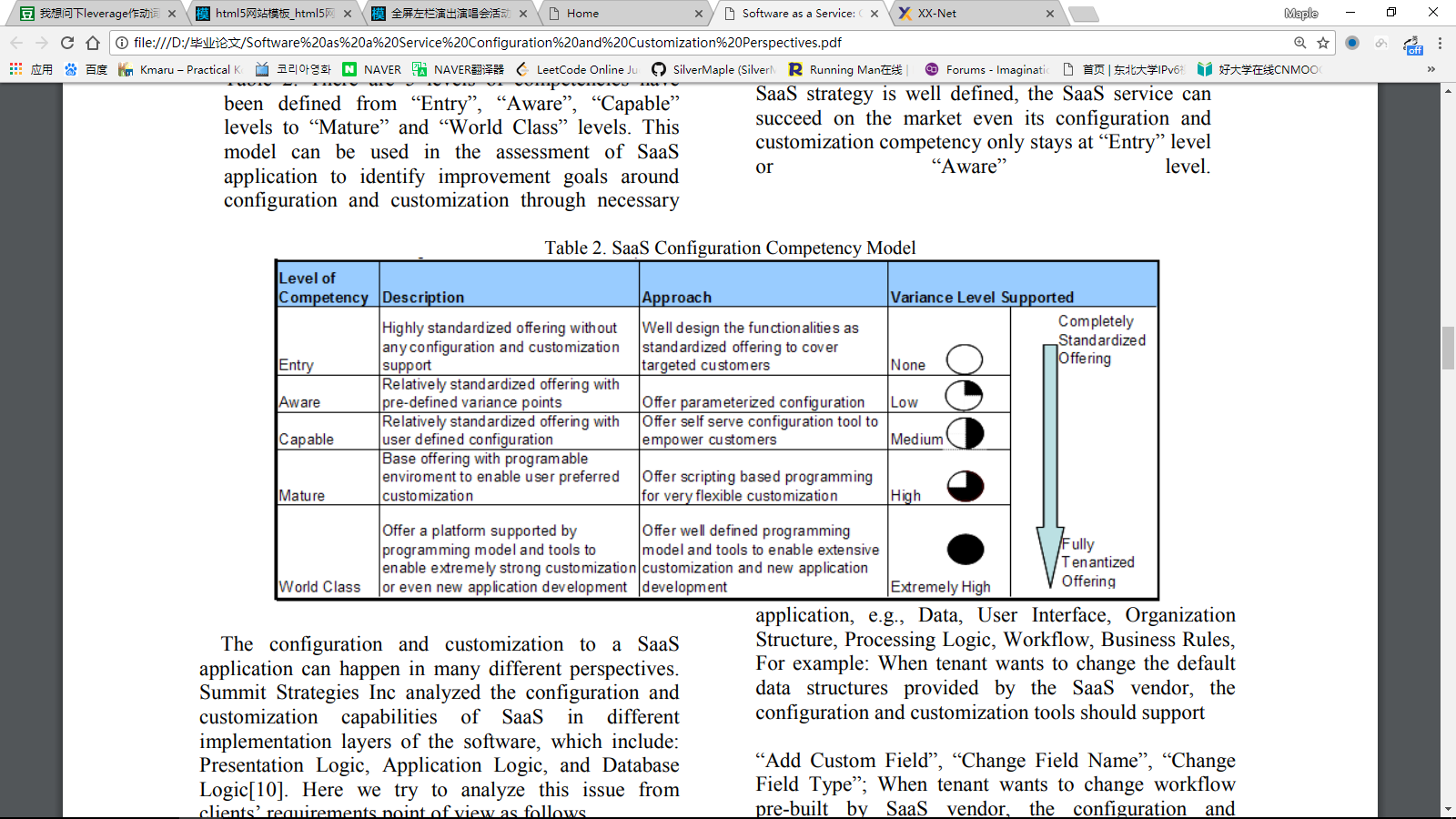


图2.配置和定制

与配置相比，定制对于saas供应商和客户来说都是一个更加昂贵的方法。因为定制涉及到saas软件源代码的变化，这产生了很多涉及显着成本的问题，例如：要求具有较高工资技能的人员进行定制;分配资源和基础设施来管理软件代码版本;涉及代码开发/调试/测试和部署带来的更长的生命周期;并且不能接受定制复杂性和成本的客户失去商业机会[9]。在saas环境下，定制变得更加复杂，因为sass供应商需要维护tenant的每一个定制代码租户。升级saas应用程序不应导致任何单个租户的定制代码丢失。因此，只要有可能，saas都应该通过使用配置来满足客户的剪裁需求，并尽可能地向客户的独特要求放大配置限制，从而避免定制。

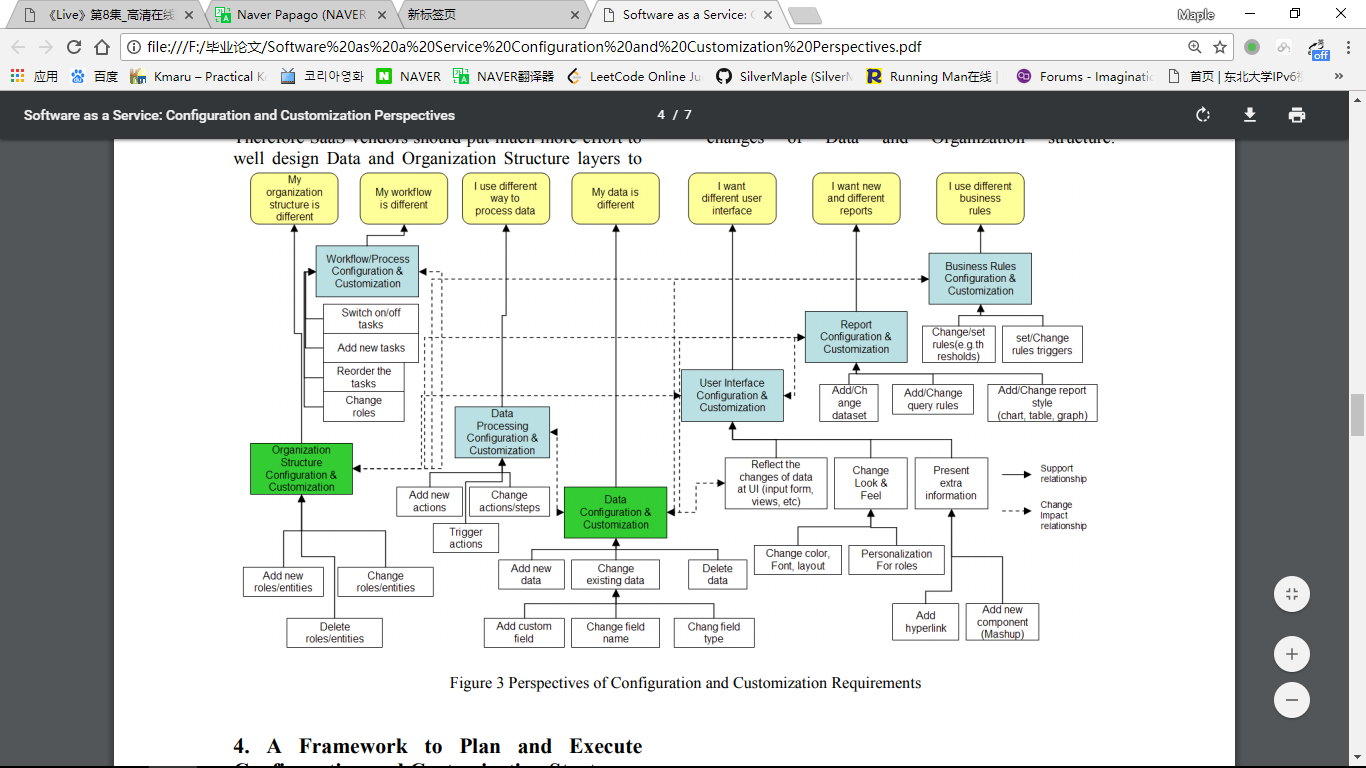
1. Configuration and Customization Competency Model

To facilitate strategy definition and execution discussion around SaaS configuration and customization, we introduce the Configuration and Customization Competency Model described in Table 2. There are 5 levels of competencies have been defined from “Entry”, “Aware”, “Capable” levels to “Mature” and “World Class” levels. This model can be used in the assessment of SaaS application to identify improvement goals around configuration and customization through necessary benchmarking with market leader’s competency level. Different level of competency can enable different level of variance requirements through different technical approaches supported by ranges of SaaS services from completely standardized offering across all the tenants to fully tenantized offering for any individual tenant. In theory, the higher of the competency level, the more customers and the more complex variance requirements the SaaS service can support. However different SaaS vendors may have different strategies in terms of targeted customer segments, supported scope of variance, etc. If the SaaS strategy is well defined, the SaaS service can succeed on the market even its configuration and customization competency only stays at “Entry” level or “Aware” level.。



The configuration and customization to a SaaS application can happen in many different perspectives. Summit Strategies Inc analyzed the configuration and customization capabilities of SaaS in different implementation layers of the software, which include: Presentation Logic, Application Logic, and Database Logic[10]. Here we try to analyze this issue from clients’ requirements point of view as follows.

As illustrated by the figure 3, SaaS tenants can potentially have configuration and customization requirements from many different perspectives. Each tenant may raise the following challenges to the SaaS vendor: “I need more fields to describe my business documents”; “Our manager wants a new report/dashboard to analyze sales data”; “Our organization has no role of procurement manager”; “The workflow of our business is different with what you can support”. Any of these challenges can be divided into implications to different perspectives of the SaaS application, e.g., Data, User Interface, Organization Structure, Processing Logic, Workflow, Business Rules, For example: When tenant wants to change the default data structures provided by the SaaS vendor, the configuration and customization tools should support “Add Custom Field”, “Change Field Name”, “Change Field Type”; When tenant wants to change workflow pre-built by SaaS vendor, the configuration and customization tools should support “Switch on/off Tasks”, “Add New Tasks”, “Reorder the Tasks”, “Change Roles for a Task”. If you analyze those change impact relationship” lines on the figure, you will notice “Data Configuration and Customization” and “Organization Structure Configuration and Customization” are the two most important perspectives, any change of these two perspectives will potentially bring major impact to many other perspectives including User Interface, Workflow, Business Rules, etc. For example: If tenant changes a data structure, then the user interface to support the input and view of the data should be changed as well; If tenant changes the roles’ definition, then the workflow need to be changed accordingly for those tasks handled by those roles. Therefore SaaS vendors should put much more effort to well design Data and Organization Structure layers to support easy configuration and customization. It is also very important to consider the impacts and establish the linkages between the other software artifacts and the changes of Data and Organization structure.



1. 配置和定制能力模型，

以便于围绕saas配置和定制进行战略定义和执行讨论，我们介绍了表2中描述的配置和定制能力模型。从“入门”，“了解”，“有能力”的水平到“成熟”和“世界级”水平。该模型可以用于评估saas应用程序，以便通过与市场领导者的能力水平进行必要的基准确定配置和定制的改进目标。不同的能力水平可以通过不同的技术方法支持不同的技术方法，从所有租户的完全标准化服务到完全为任何单个租户提供租赁服务。从理论上讲，能力水平越高，客户越多，变化级别要求越复杂，saas服务可以支持。然而，不同的Saas供应商可能在目标客户群，支持差异范围等方面有不同的策略，如果Saas战略定义得当，saas服务可以在市场上取得成功，即使其配置和定制能力只停留在“入门”水平或“了解”水平。

表2. SaaS配置能力模型

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 能力级别 | 描述 | 方法 | 支持的变化级别 | |
| 入门 | 高度标准化的产品，无需任何配置和定制支持 | 精心设计功能作为标准服务，以覆盖目标客户 | 无 | 完全标准化产品  完全租户化产品 |
| 了解 | 有预定义变化点的相对标准化产品 | 提高参数化配置 | 低 |
| 有能力 | 有用户定义配置的相对标准化产品 | 提供自服务配置工具来赋予客户权力 | 中 |
| 成熟 | 具有可编程环境的基础标准化产品，支持用户定制 | 为非常灵活的定制提供基于脚本的编程 | 高 |
| 世界级 | 提供一个由编程模型和工具支持的平台，以实现极强的定制甚至是新的应用程序开发 | 提供定义良好的编程模型和工具，以实现广泛的定制和新应用程序开发 | 极高 |

saas应用程序的配置和定制可以在许多不同的角度进行。峰会策略公司分析了软件不同实现层面上saas的配置和定制功能，其中包括：表示逻辑，应用逻辑和数据库逻辑[10]。在这里我们试图从客户的要求角度来分析这个问题，如下所示。

如图3所示，saas租户可能会有许多不同角度的配置和定制需求。每个租户都可能向SaaS供应商提出以下挑战：“我需要更多的领域来描述我的商业文件”;“我们的经理想要一个新的报告/仪表板来分析销售数据”;“我们的组织没有采购经理的角色”;“我们业务的工作流程与您所能支持的不同”。这些挑战中的任何一个都可以分为对saas应用的不同观点的影响，例如数据，用户界面，组织结构，处理逻辑，工作流程，业务规则，例如：当租户想要更改由saas供应商提供的默认数据结构时，配置和定制工具应该支持“添加自定义字段”，“更改字段名称”，“更改字段类型”;当租户想要更改由saas供应商预先构建的工作流程时，配置和自定义工具应该支持“开关任务”，“添加新任务”，“重新排序任务”，“更改任务角色”。如果分析图中的这些变化影响关系“线，​​您会注意到”数据配置和定制“和”组织结构配置和定制“是两个最重要的视角，这两个视角的任何变化都可能对许多人产生重大影响其他观点，包括用户界面，工作流程，业务规则等. 例如：如果租户更改数据结构，则应该更改用于支持数据输入和查看的用户界面;如果租户更改角色的定义，那么工作流程需要相应地更改由这些角色处理的任务。因此saas供应商应该更加努力地设计数据和组织结构层以支持简单的配置和定制。考虑影响并建立其他软件产品与数据和组织结构变化之间的联系也非常重要。

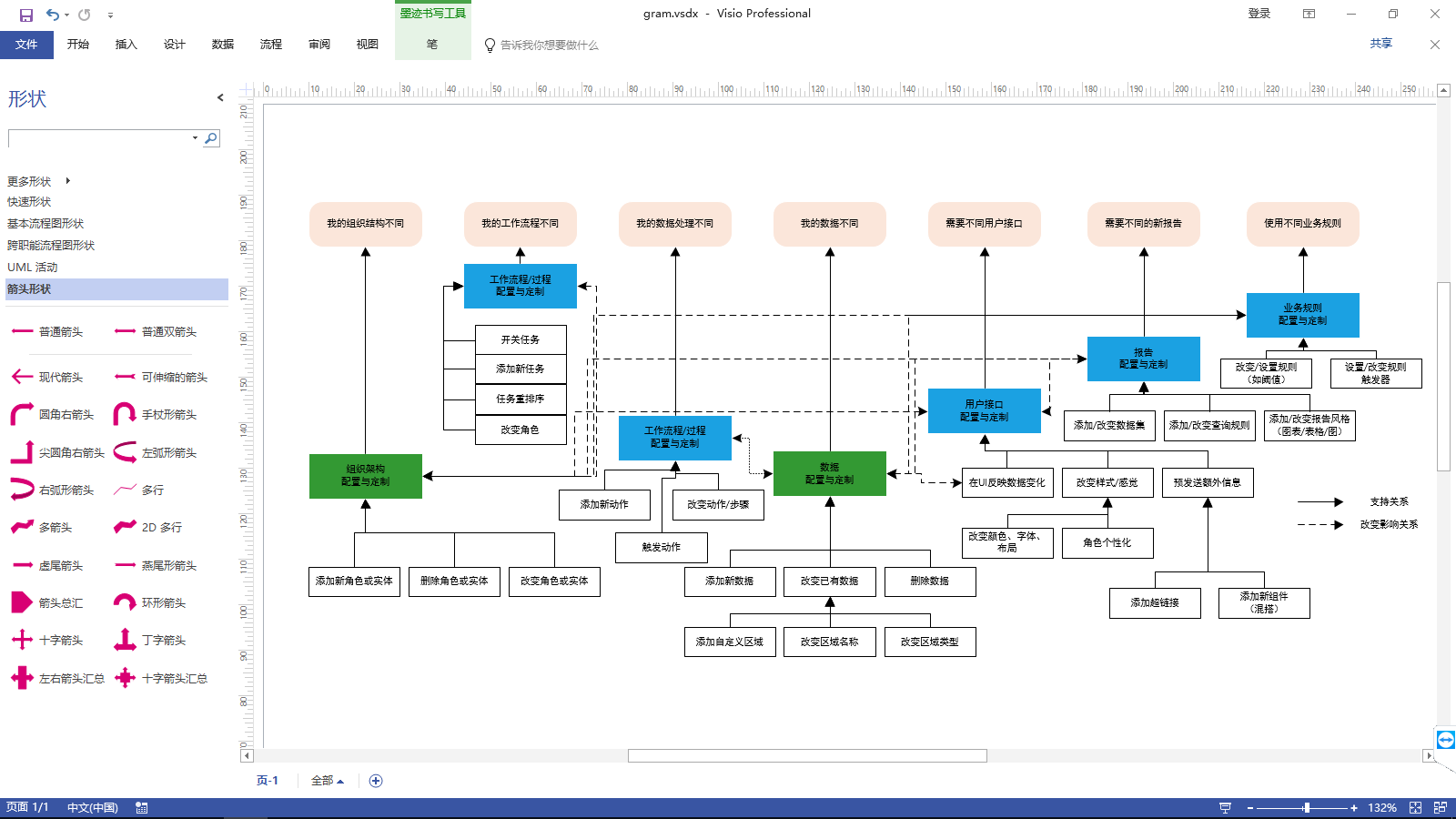
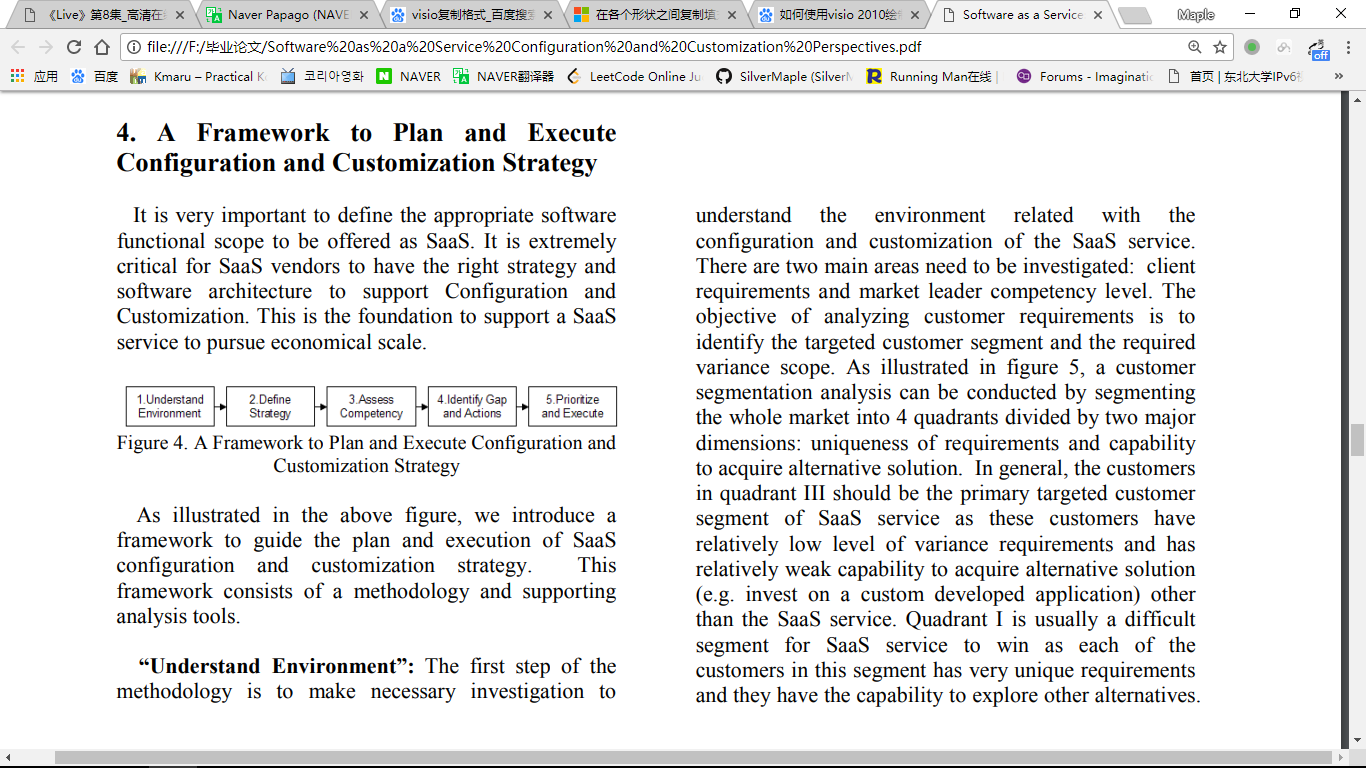


图3 配置和定制要求透视图

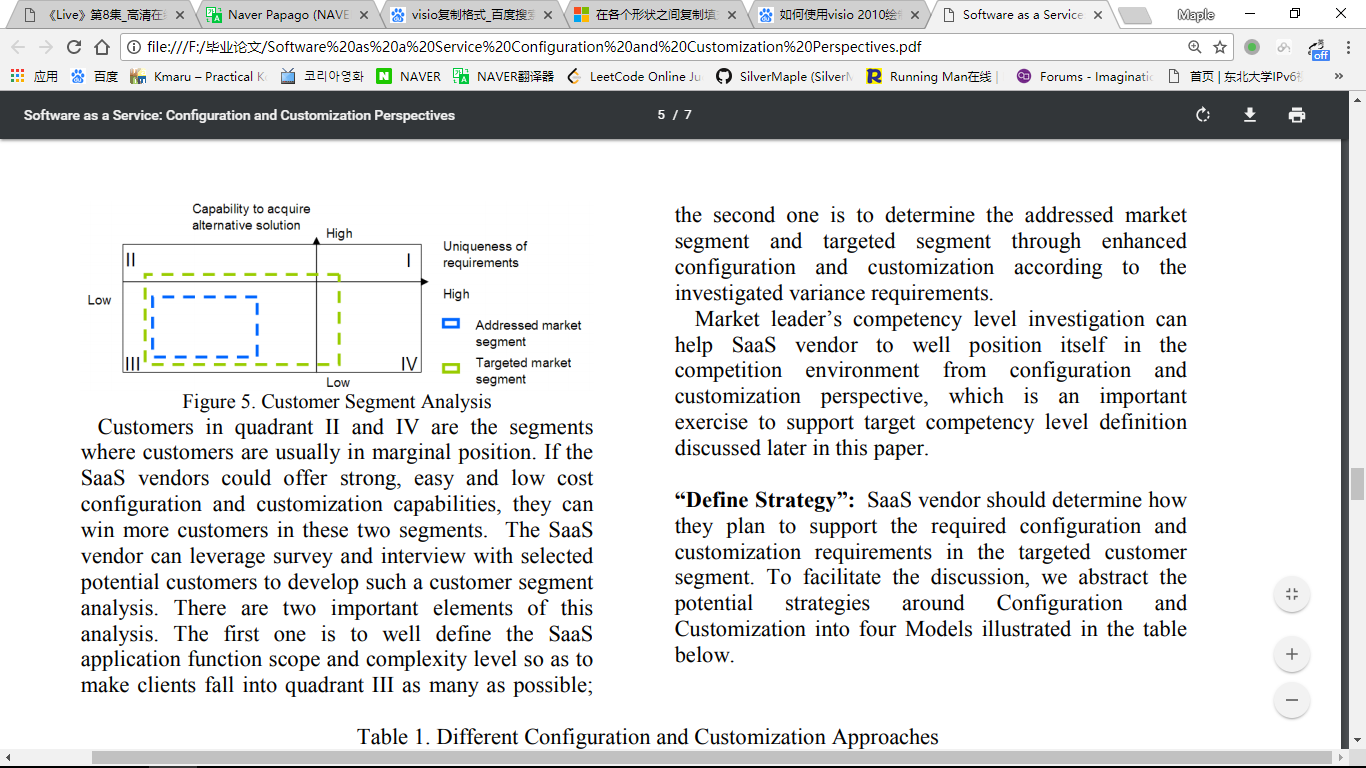
1. A Framework to Plan and Execute Configuration and Customization Strategy

It is very important to define the appropriate software functional scope to be offered as SaaS. It is extremely critical for SaaS vendors to have the right strategy and software architecture to support Configuration and Customization. This is the foundation to support a SaaS service to pursue economical scale.



As illustrated in the above figure, we introduce a framework to guide the plan and execution of SaaS configuration and customization strategy. This framework consists of a methodology and supporting analysis tools.

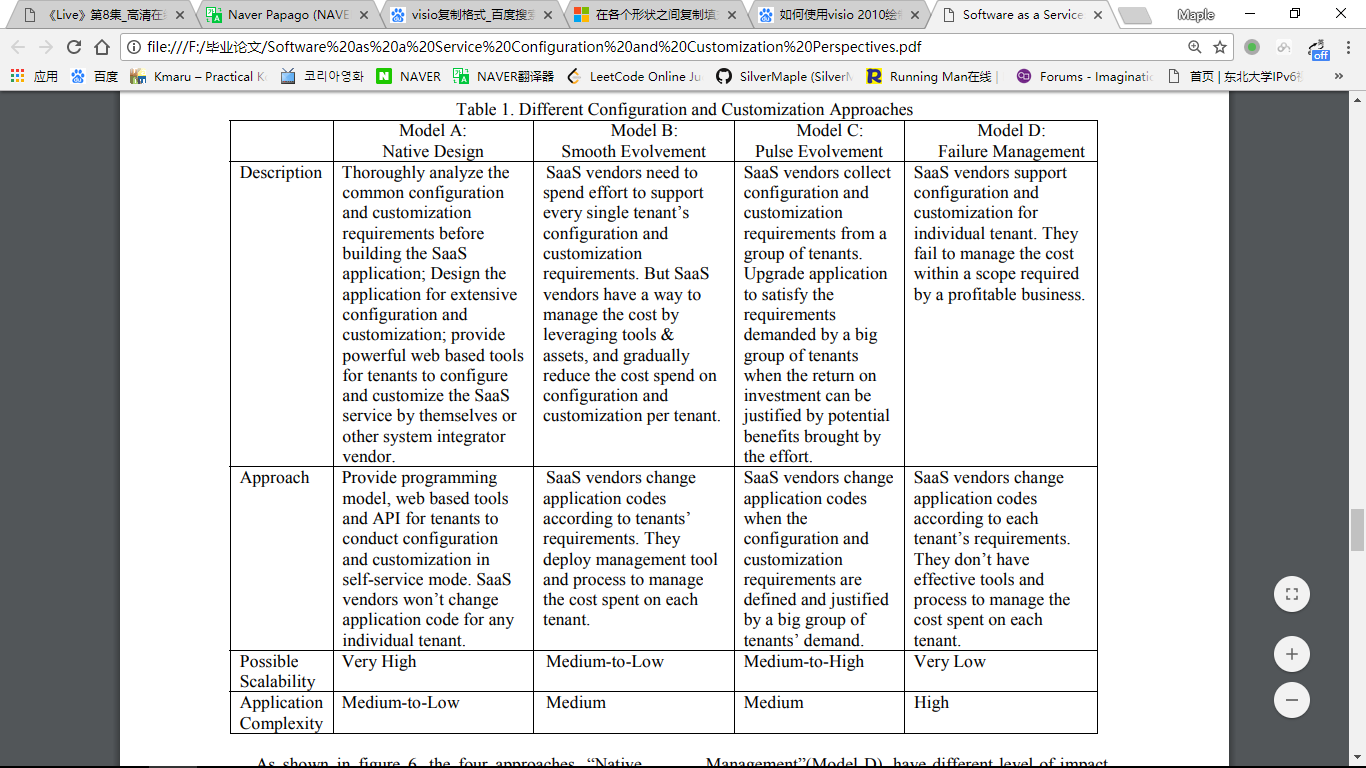
“Understand Environment”: The first step of the methodology is to make necessary investigation to understand the environment related with the configuration and customization of the SaaS service. There are two main areas need to be investigated: client requirements and market leader competency level. The objective of analyzing customer requirements is to identify the targeted customer segment and the required variance scope. As illustrated in figure 5, a customer segmentation analysis can be conducted by segmenting the whole market into 4 quadrants divided by two major dimensions: uniqueness of requirements and capability to acquire alternative solution. In general, the customers in quadrant III should be the primary targeted customer segment of SaaS service as these customers have relatively low level of variance requirements and has relatively weak capability to acquire alternative solution (e.g. invest on a custom developed application) other than the SaaS service. Quadrant I is usually a difficult segment for SaaS service to win as each of the customers in this segment has very unique requirements and they have the capability to explore other alternatives.



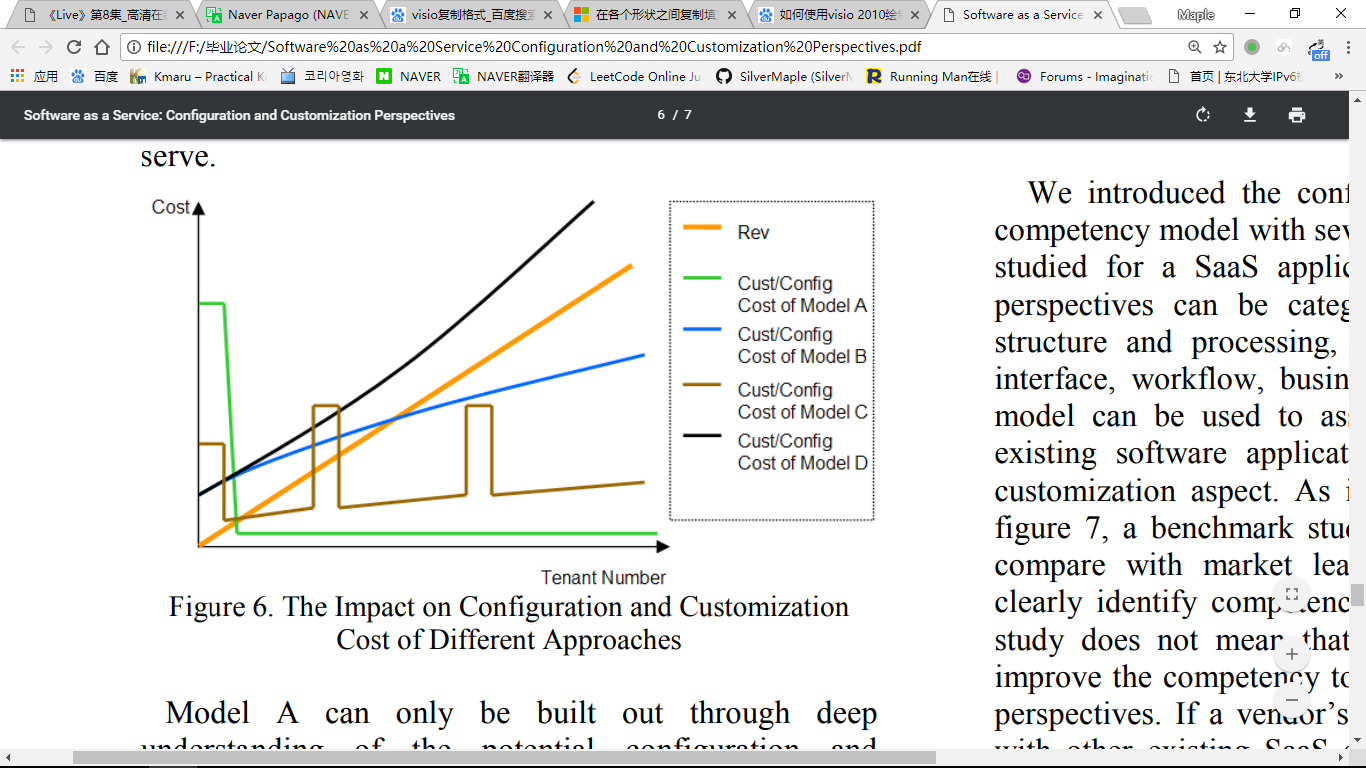
Customers in quadrant II and IV are the segments where customers are usually in marginal position. If the SaaS vendors could offer strong, easy and low cost configuration and customization capabilities, they can win more customers in these two segments. The SaaS vendor can leverage survey and interview with selected potential customers to develop such a customer segment analysis. There are two important elements of this analysis. The first one is to well define the SaaS application function scope and complexity level so as to make clients fall into quadrant III as many as possible; the second one is to determine the addressed market segment and targeted segment through enhanced configuration and customization according to the investigated variance requirements.

Market leader’s competency level investigation can help SaaS vendor to well position itself in the competition environment from configuration and customization perspective, which is an important exercise to support target competency level definition discussed later in this paper.

“Define Strategy”: SaaS vendor should determine how they plan to support the required configuration and customization requirements in the targeted customer segment. To facilitate the discussion, we abstract the potential strategies around Configuration and Customization into four Models illustrated in the table below.



As shown in figure 6, the four approaches, “Native Design”(Model A), “Smooth Evolvement”(Model B), “Pulse Evolvement”(Model C), “Failure Management”(Model D), have different level of impact on the SaaS service delivery cost spent on each tenant from configuration and customization. As shown on the following figure, Model D is obviously a bad one which every SaaS vendors should avoid to get into. The other three models can all support sustainable SaaS service business with different profit margins. They can be good choice according to specific SaaS business context in terms of: application complexity, scalability target, the vendor’s understanding of the market, the budget situation, etc. In general, Model B is more appropriate for SaaS targeting very limited number of clients as supporting each individual tenant’s unique requirements is a very expensive strategy. If a SaaS vendors want to explore very high scalability to leverage very big economical scale (Long tail play), then Model A would be the best approach. The easiest approach for a SaaS vendor starts from is Model C, they learn the market along the process and eventually can be evolved into Model A when they clearly define and build configuration and customization capabilities needed by the large amount of tenants they want to acquire and serve.

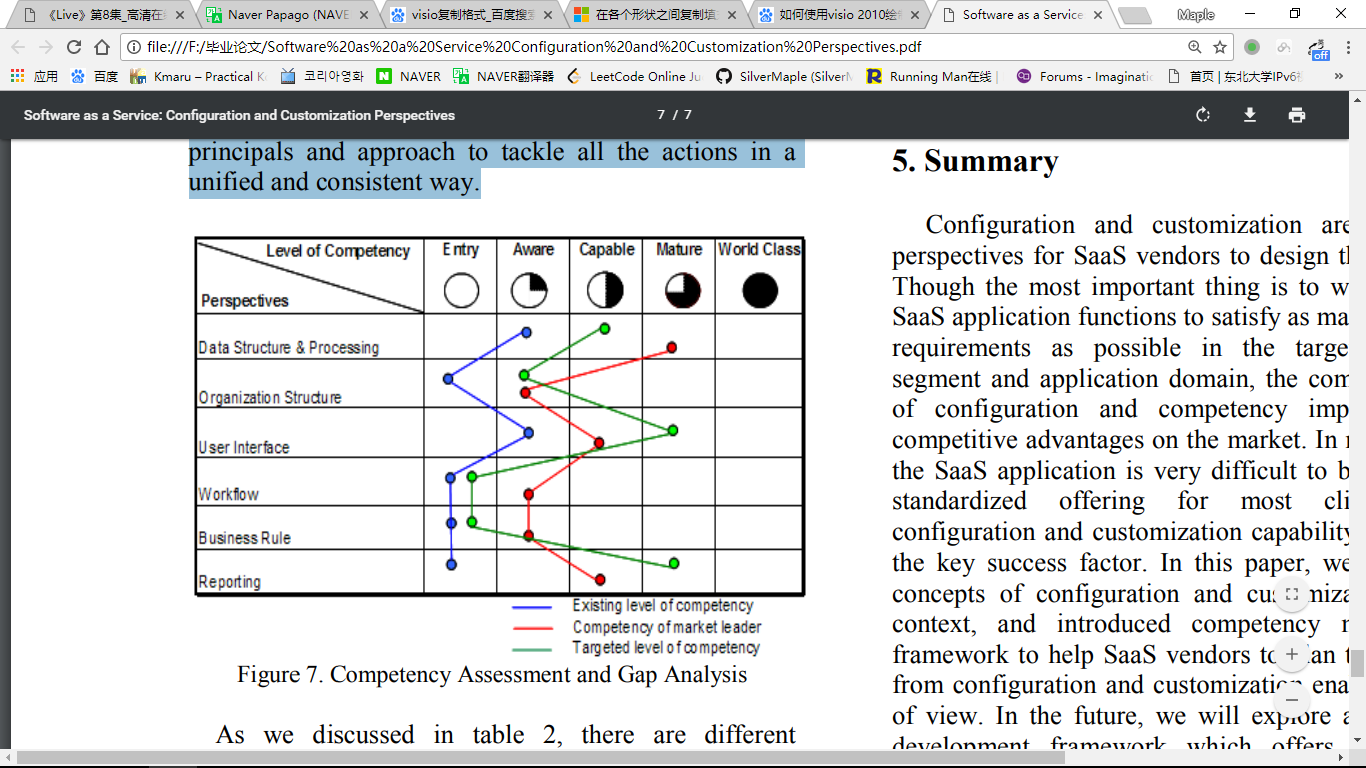


Model A can only be built out through deep understanding of the potential configuration and customization requirements associated with the SaaS service. It takes specially designed software architecture and provides web based tools for easy and extensive configuration and customization without changing the SaaS application source code.

“Assess Competency”: This step is extremely important for those traditional software application vendors who plan to explore SaaS as a new delivery model. The configuration and customization might not be a big challenge for applications vendors who have been successfully addressing consumer market and small medium business(SMB) market. These vendors usually take volume play model and do not support individual customer’s variance requirements. They can jump start to explore SaaS by transforming their applications into multi-tenancy enabled with Web interface. However the configuration and customization issue is a big challenge for those application vendors who have been addressing medium to large enterprise market. Though they have well packaged application as a base, these vendors are usually paid by their customers to take custom development approach to satisfy each individual customer’s unique variance requirements. In many cases, source code level customizations are involved if the application has no well defined configuration framework. But in the traditional application delivery model, the vendor can afford that because they are paid by the end customer to do so. This approach can not be replicated in SaaS delivery model. SaaS has subscription based usage pattern. The very small amount upfront investment made by the tenant and monthly based subscription fee can not support the total cost spent on source code level customization. Therefore these application vendors should be very careful and make necessary assessment about their competency around configuration and customization before they decide to move their application to the SaaS delivery model.

We introduced the configuration and customization competency model with several major perspectives to be studied for a SaaS application in section 3. These perspectives can be categorized into 6 groups: data structure and processing, organization structure, user interface, workflow, business rule and reporting. This model can be used to assess the competency of the existing software application from configuration and customization aspect. As illustrated by an example in figure 7, a benchmark study can also be conducted to compare with market leader’s competency so as to clearly identify competency improvement goals. This study does not mean that every SaaS vendor should improve the competency to the higher level from every perspectives. If a vendor’s application is pretty similar with other existing SaaS services on the market from function aspect, then higher level configuration and customization competency can help the vendor to get stronger competitive advantages.

“Identify Gaps and Actions”: Through the competency benchmark study, the competency gaps can be identified to guide the actions’ definition. The example in figure 7 shows the gaps and improvement goals especially in user interface and reporting perspectives. With the analysis in section 3, the competency improvement goal could be further developed into specific actions. For example, to improve the configuration and customization capability for reporting function of the application, the actions need to be identified to support “Add/Change dataset”, “Add/Change query rules” and “Add/Change report style (chart, table, graph)”. If we go through every configuration and customization perspectives according to figure 3, we can generate a long list of actions which implies very complex design challenges for the SaaS application. It is critical to have well designed principals and approach to tackle all the actions in a unified and consistent way.



As we discussed in table 2, there are different approaches which can enable different competency level requirements. Parameterized Configuration can enable “Aware” level variance through setting pre-defined parameters and options by end user in the runtime environment. Self Serve Configuration tool leverages an application variance metadata framework and a series of simple point-and-click wizards, users can design custom user interfaces and modify the structure of the data model and the application’s business logic (workflow, business rules, etc). But the scope of configuration is constrained by the metadata framework. Scripting based programming, a version of end-user programming, allow for larger scope customization by the end user by extending the features of the tool using a constraint scripting language to guarantee security and avoid script generated damage to the core application. World Class SaaS service make their application coupled with a development environment and a formal programming model that can be used by user to build new application code or modify compiled code to match their requirements. Different SaaS vendors take different approach and develop their own implementations.[11] There is not a generic and platform independent approach for SaaS vendors to use today. There is a strong opportunity for research activities.

In this section, we introduced a framework to guide SaaS vendors to plan and execute configuration and customization strategy. This framework comes from our hands on experience and lesson learned through developing SaaS application and business. The framework can be tailored to help SaaS customers to evaluate different SaaS vendors to make the subscription decision.

1. 计划和执行配置和定制策略的框架

定义适当的软件功能范围作为saas提供服务是非常重要的。对于Saas供应商来说，拥有正确的战略和软件架构来支持配置和定制是非常关键的。这是支持saas服务追求经济规模的基础。

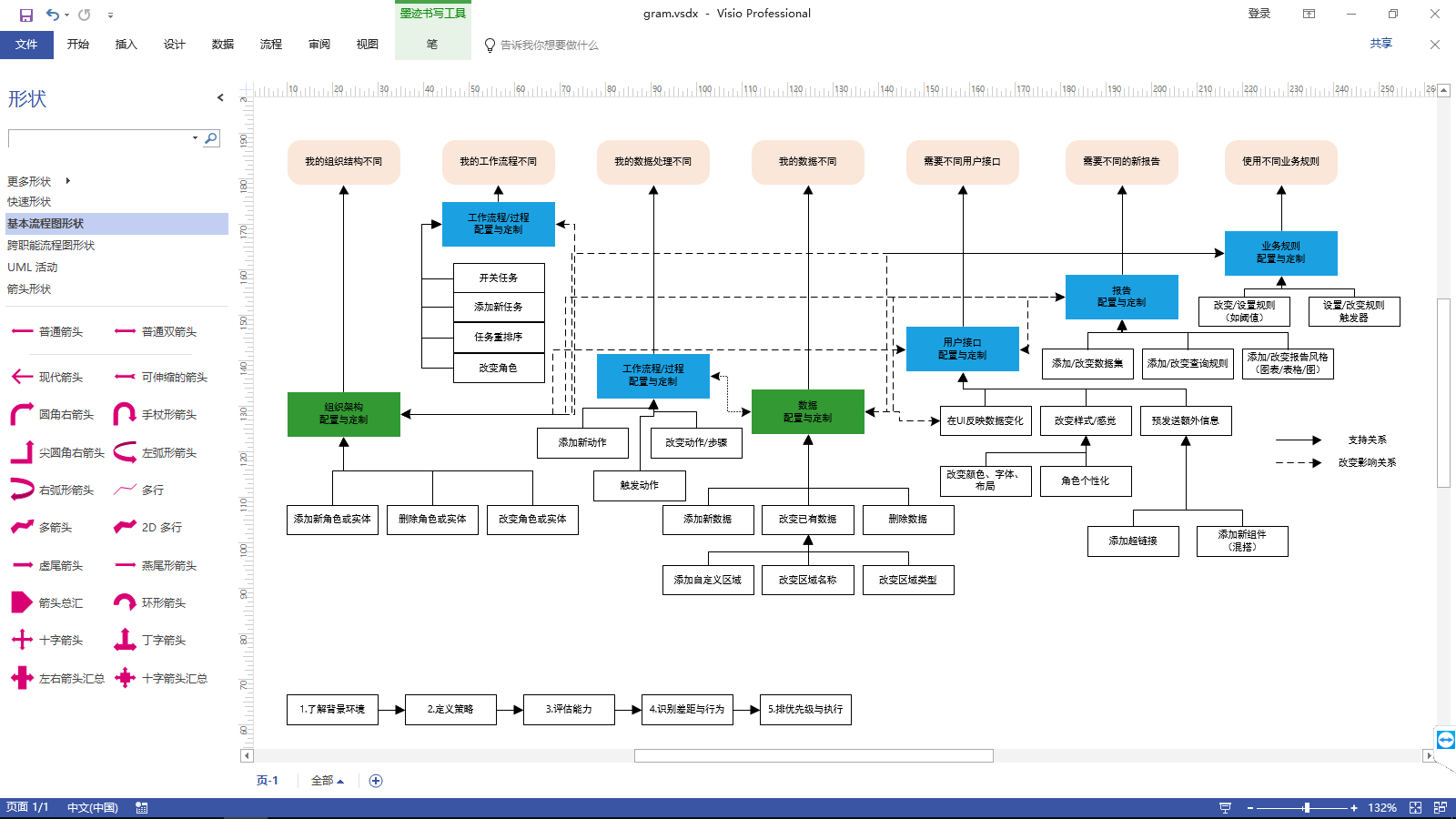


图4. saas配置和定制策略的计划和执行框架

如上图所示，我们引入了一个框架来指导saas配置和定制策略的计划和执行。这个框架由一个方法和支持分析工具组成。

“了解背景环境”：方法的第一步是进行必要的调查，以了解与saas服务的配置和定制有关的环境。有两个主要领域需要调查：客户需求和市场领导者能力水平。分析客户需求的目标是确定目标客户群和所需的差异范围。如图5所示，客户细分分析可以通过将整个市场细分为4个象限，除以两个主要维度：要求的唯一性和获得替代解决方案的能力。一般来说，第三象限中的客户应该是SaaS服务的主要目标客户群，因为这些客户的差异化要求水平相对较低，并且相对较弱地获得除saas之外的替代解决方案（例如投资于定制开发的应用程序）的能力服务。象限i通常是SaaS服务赢得的困难部分，因为该部门的每个客户都有非常独特的要求，并且他们有能力探索其他替代方案。

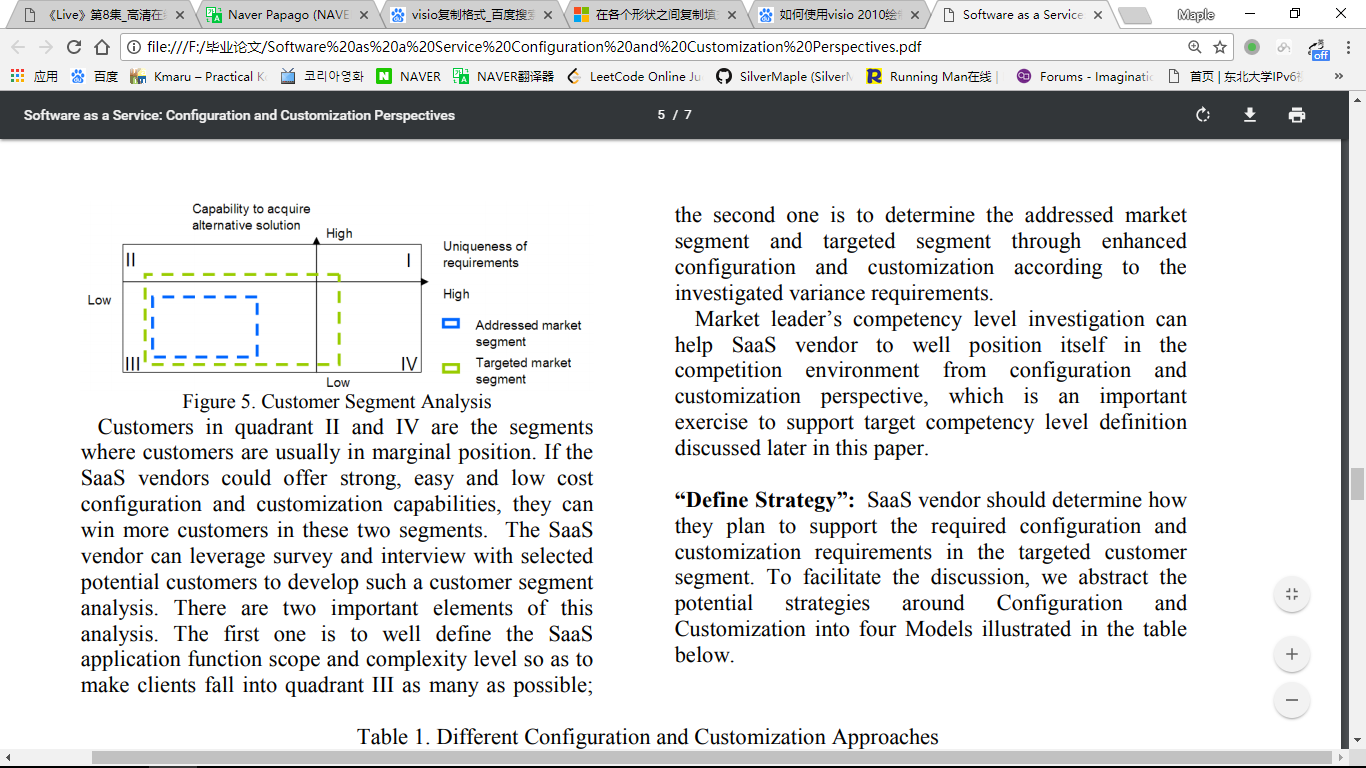


图5. 客户群分析

第二象限和第四象限中的顾客通常处于边缘位置。如果SaaS供应商可以提供强大，简单和低成本的配置和定制功能，他们可以在这两个领域赢得更多的客户。SaaS供应商可以利用对选定潜在客户的调查和访谈来开发此类客户细分分析。这个分析有两个重要的元素。首先是为了使客户尽可能多地进入第三象限，对Saas应用功能范围和复杂程度进行定义;第二个是根据调查的变动要求，通过增强的配置和定制来确定所处理的细分市场和目标细分市场。

市场领导者的能力水平调查可以帮助Saas供应商从配置和定制角度很好地将自己置于竞争环境中，这是本文后面讨论的支持目标胜任能级定义的重要练习。

“定义战略”：saas供应商应确定他们计划如何支持目标客户群中所需的配置和定制要求。为了便于讨论，我们将围绕配置和定制的潜在战略抽象为下表所示的四个模型。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 模型A:  本地设计 | 模型B:  平滑演变 | 模型C:  脉冲演变 | 模型D:  失效管理 |
| 描述 | 在构建saas应用程序之前，深入分析常见的配置和定制要求;为广泛的配置和定制设计应用程序;为租户提供强大的基于Web的工具，以便自己或其他系统集成商供应商配置和定制saas服务。 | Saas供应商需要花费精力来支持每个租户的配置和定制需求。但是萨斯供应商有办法通过利用工具和资产来管理成本，并逐渐减少每个租户的配置和定制成本。 | Saas供应商收集来自一组租户的配置和定制要求。升级应用程序以满足一大群租户要求的投资回报可以通过努力带来的潜在利益。 | Saas供应商支持个人租户的配置和定制。他们无法在盈利业务所要求的范围内管理成本。 |
| 方法 | 为租户提供编程模型，基于web的工具和api以自助服务模式进行配置和定制。saas供应商不会为任何个人租户更改应用程序代码。 | Saas供应商根据租户的要求更改应用程序代码。他们部署管理工具和流程来管理每个租户花费的成本。 | Saas供应商在配置和定制需求被定义并且由一大群租户的需求所证明的情况下更改应用程序代码。 | Saas供应商根据每个租户的要求更改应用程序代码。他们没有有效的工具和流程来管理每个租户花费的成本。 |
| 可能的可扩展性 | 非常高 | 中至低 | 中至高 | 非常低 |
| 应用复杂性 | 中至低 | 中 | 中 | 高 |

如图6所示，“本地设计”（模型a），“平滑演变”（模型b），“脉冲演变”（模型c），“失效管理”（模型d）这四种方法具有不同程度的影响从配置和定制花费在每个租户上的SaaS服务交付成本。如下图所示，模型d显然是一个糟糕的问题，每个saas供应商都应该避免进入。其他三种模式都可以支持不同利润率的可持续的SaaS服务业务。根据特定的saas业务环境，它们可以是很好的选择：应用程序复杂性，可扩展性目标，供应商对市场的了解，预算情况等，一般而言，b型更适合saas，目标客户数量非常有限因为支持每个租户的独特需求是一项非常昂贵的策略。如果一个SaaS供应商想要探索非常高的可扩展性以利用非常大的经济规模（长尾巴），那么模型a将是最好的方法。saas供应商最简单的方法是从模型c开始，他们沿着流程学习市场，并最终在他们明确定义和构建他们想要获取和服务的大量租户所需的配置和定制功能时演变成模型a。

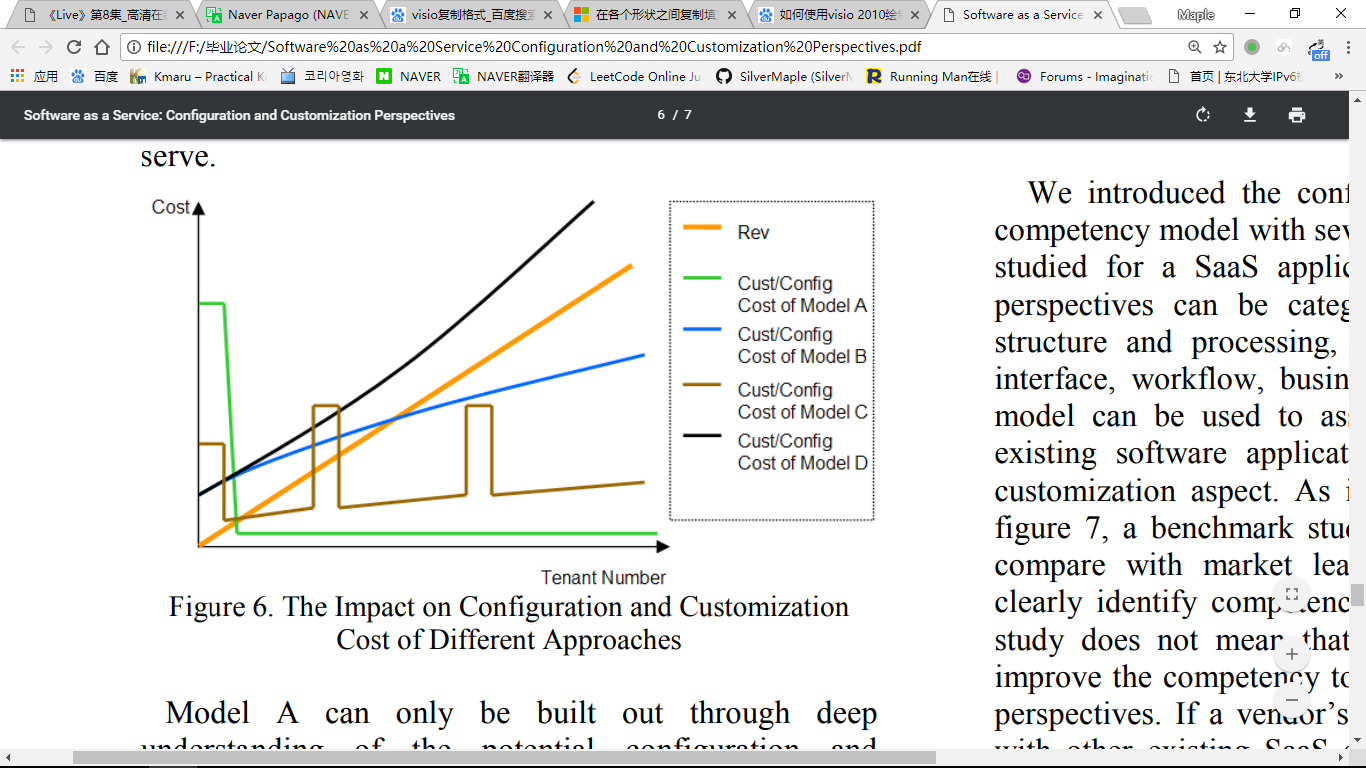


图6. 不同方法对配置和定制成本的影响

模型a只能通过深入了解与saas服务相关的潜在配置和定制需求才能构建出来。它需要特别设计的软件架构，并提供基于Web的工具，以便进行简单而广泛的配置和定制，而无需更改saas应用程序源代码。

“评估能力”：对于那些计划将saas作为新的交付模式进行探索的传统软件应用供应商来说，这一步非常重要。对于成功解决消费者市场和中小型企业（smb）市场的应用程序供应商来说，配置和定制可能不是一个大的挑战。这些供应商通常采用批量销售模式，不支持个别客户的差异要求。他们可以通过将应用程序转换为支持Web界面的多租户来开始探索saas。然而，对于那些一直致力于大中型企业市场的应用程序供应商来说，配置和定制问题是一个巨大的挑战。尽管他们已经将打包应用作为基础，但这些供应商通常由客户支付费用，以采用定制开发方法来满足每个客户的独特差异需求。在许多情况下，如果应用程序没有定义良好的配置框架，则会涉及源代码级别自定义。但在传统的应用交付模式中，供应商可以负担得起，因为他们是由最终用户支付的。这种方法无法在saas交付模式中复制。saas有基于订阅的使用模式。由租户进行的非常少量的前期投资和按月订购费用不能支持在源代码级别定制上花费的总成本。因此这些应用程序供应商应该非常小心，并在他们决定将应用程序迁移到saas交付模式之前对其配置和定制能力进行必要的评估。

我们在第3节中介绍了配置和定制能力模型，其中包含几个主要的视角来研究saas应用。这些视角可以分为6组：数据结构和处理，组织结构，用户界面，工作流程，业务规则和报告。此模型可用于从配置和定制方面评估现有软件应用程序的能力。如图7中的例子所示，还可以进行基准研究，以与市场领导者的能力进行比较，以清楚地识别能力提升目标。这项研究并不意味着每个SaaS供应商都应该从各个角度将能力提高到更高的水平。如果供应商的应用程序与功能方面的市场上其他现有的SaaS服务非常相似，那么更高级别的配置和定制能力可以帮助供应商获得更强大的竞争优势。

“识别差距和行为”：通过能力基准研究，可以识别能力差距来指导行为的定义。图7中的例子显示了差距和改进目标，尤其是在用户界面和报告视角方面。通过第3节的分析，能力提升目标可以进一步发展为具体行动。例如，为了提高应用程序报告功能的配置和定制能力，需要确定动作以支持“添加/更改数据集”，“添加/更改查询规则”和“添加/更改报告样式（图表，表格，图）“。如果我们按照图3通过每个配置和定制视角，我们可以生成一长串行动，这意味着saas应用程序的复杂设计挑战。设计良好的校长和方法以统一和一致的方式处理所有行动至关重要。

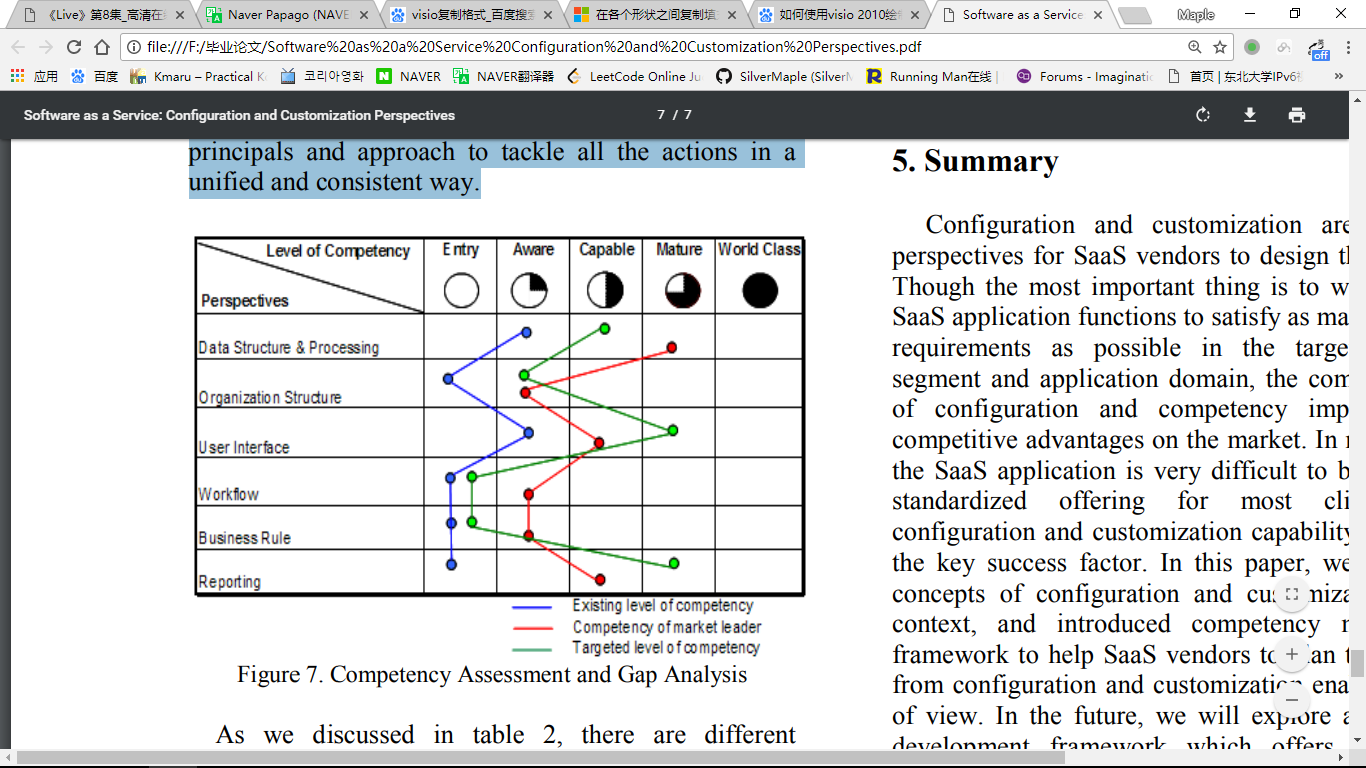


图7.能力评估和差距分析

正如我们在表2中讨论的那样，有不同的方法可以实现不同的能力水平要求。参数化配置可以通过在运行时环境中由最终用户设置预定义的参数和选项来启用“感知”水平差异。自助配置工具利用应用程序差异元数据框架和一系列简单的点击式向导，用户可以设计自定义用户界面并修改数据模型的结构和应用程序的业务逻辑（工作流，业务规则等）。但配置的范围受到元数据框架的限制。基于脚本的编程是最终用户编程的一个版本，允许最终用户通过使用约束脚本语言扩展工具的功能来实现更大范围的自定义，从而保证安全性并避免脚本对核心应用程序造成损害。世界级的Saas服务使其应用程序与开发环境和正式编程模型相结合，用户可以使用它们构建新的应用程序代码或修改编译后的代码以符合其要求。不同的Saas供应商采取不同的方法并开发他们自己的实现。[11]目前Saas供应商没有一个通用且平台独立的方法。这是一个很好的研究活动的机会。

在本节中，我们引入了一个框架来指导saas供应商计划和执行配置和定制策略。这个框架来自我们通过开发saas应用程序和业务的经验和教训。该框架可以定制，以帮助saas客户评估不同的saas供应商以作出订购决定。

1. Summary

Configuration and customization are the critical perspectives for SaaS vendors to design their offerings. Though the most important thing is to well design the SaaS application functions to satisfy as many customers’ requirements as possible in the targeted customer segment and application domain, the competency level of configuration and competency implies the key competitive advantages on the market. In many cases, if the SaaS application is very difficult to be designed as standardized offering for most clients, strong configuration and customization capability will become the key success factor. In this paper, we clarified the concepts of configuration and customization in SaaS context, and introduced competency model and a framework to help SaaS vendors to plan their offerings from configuration and customization enablement point of view. In the future, we will explore a runtime and development framework which offers programming model and environment to enable SaaS vendors to quickly build highly configurable and customizable SaaS applications.

1. 总结

配置和定制是SaaS供应商设计产品的关键视角。尽管最重要的是要良好设计saas应用功能以满足目标客户群和应用领域中尽可能多的客户需求，但配置和能力的能力水平意味着市场上的关键竞争优势。在很多情况下，如果saas应用程序很难被设计为大多数客户的标准化产品，那么强大的配置和定制功能将成为关键的成功因素。在本文中，我们阐明了saas背景下的配置和定制概念，并介绍了能力模型和框架，以帮助saas供应商从配置和定制支持角度来规划他们的产品。未来，我们将探索一个运行时和开发框架，它提供了编程模型和环境，使saas供应商能够快速构建高度可配置和可定制的saas应用程序。

References

[1] E. Knorr, “Software as a Service: The Next Big

Thing”,http://www.infoworld.com/article/06/03/20/7610

3\_12FEsaas\_1.html.

[2]Wei Sun, etc, “Software as Service: An Integration

Perspective”, ICSOC 2007.

[3] Chang jie Guo, etc, “A Framework for Native MultiTenancy

Application Development and Management”,

CEC/EEE 2007.

[4] Sheryl Kingstone, “Understanding Total Cost of

Ownership of a Hosted vs. Premises-Based CRM

Solution”, Yankee Group Report June 2004.

[5] Roger Pressman, “Software Engineering: A

Practitioner's Approach”, McGraw-Hill Science.

[6] SAP, “Getting Started: ABAP, SAP Community

Network

[7] Salesforce.com, “Introduction to the Apex Platform”

[8] Summit Strategy, “Software-Powered Services NetNative

Software-as-Services Transforms the ISV

Business Model”

[9] Clotilde Rohleder, “SOFTWARE

CUSTOMIZATIONWITH XML”, Volume VI, No. 2,

2005 , Issues in Information Systems.

[10] Summit Strategy, “Software Powered Services Net

Native SaS Transforms The Enterprise Application”

[11] Datamonitor ComputerWire, “SaaS Customization”