Trade-off Service Portfolio Planning – A Case Study on Mining the Android App Market

Service portfolio planning is the process of designing collections of services and deciding on their provision. The problem is highly information intensive, and most of the information required is hard to gather. In this paper, we present a solution approach based on the paradigm of Analytical Open Innovation (AOI). Open innovation is a cheap and low risk problem solving approach which relies on knowledge exchange with outside of company as a competitive advantage. Different forms of open innovation; crowd source, open source and outsource; facilitate the provider and consumer interactions and brings higher customer value. In our proposed AOI approach, open innovation is utilized for elicitation of services from web data, crowdsourcing the service value from potential customers and for the estimation of service implementation effort. For service evaluation, we apply the Kano theory of product development and customer satisfaction. Based on that and as the result from an optimization process, resource-optimized service portfolios are created that constitute trade-offs in balancing between gained value and effort needed. As a proof of concept, the proposed approach is illustrated via a case study project for the composition of Over the Top TV (OTT) services. The atomic services from 241 qualified apps were analyzed from the android app market. We demonstrate that the proposed approach is able to generate optimized trade-off solutions, composing better apps at each capacity level and achieving better customer satisfaction .The level of improvement in customer satisfaction varies between 16.5% and 95.3%.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1354v1 | CC-BY 4.0 Open Access | rec: 8 Sep 2015, publ: 8 Sep 2015

Trade-off Service Portfolio Planning – A Case Study on Mining the Android App Market

Maleknaz Nayebi & Guenther Ruhe

University of Calgary
Software Engineering Decision Support Laboratory,
University of Calgary, 2500 University Drive NW, Calgary, Alberta T2N 1N4
{mnayebi, ruhe}@ucalgary.ca

Abstract.

Service portfolio planning is the process of designing collections of services and deciding on their provision. The problem is highly information intensive, and most of the information required is hard to gather. In this paper, we present a solution approach based on the paradigm of Analytical Open Innovation (AOI). Open innovation is a cheap and low risk problem solving approach which relies on knowledge exchange with outside of company as a competitive advantage. Different forms of open innovation; crowd source, open source and outsource; facilitate the provider and consumer interactions and brings higher customer value. In our proposed AOI approach, open innovation is utilized for elicitation of services from web data, crowdsourcing the service value from potential customers and for the estimation of service implementation effort. For service evaluation, we apply the Kano theory of product development and customer satisfaction. Based on that and as the result from an optimization process, resource-optimized service portfolios are created that constitute trade-offs in balancing between gained value and effort needed. As a proof of concept, the proposed approach is illustrated via a case study project for the composition of Over the Top TV (OTT) services. The atomic services from 241 qualified apps were analyzed from the android app market. We demonstrate that the proposed approach is able to generate optimized trade-off solutions, composing better apps at each capacity level and achieving better customer satisfaction .The level of improvement in customer satisfaction varies between 16.5% and 95.3%.

Keywords: Service portfolio planning, Open Innovation, Crowdsourcing, Kano analysis, Optimization, OTT services.

1 Introduction

Services are "economic activities offered by one party to another, most commonly employing time-based performance to bring about desired results in recipients themselves, or in objects or other assets for which purchasers have responsibility" [1]. As studied in service science, management and engineering, composition of services has a broad range of applications ranging from telecommunication to health care and financial services. While services are applicable for a wide range of domains, we consider their application in IT services.

Designing service portfolios are of high importance as the combination of services needs specific information and brings higher value comparing to individual ones. Designing service portfolios is a highly decision-centric and information-intensive process mentioned to address when? What and How good? Service portfolio planning is increasingly under pressure to adapt to the radically changing business conditions. As a consequence, the related decision processes also need to be adapted. The reactive mode of operation needs to be replaced by real-time or even pro-active decisions. The information these decisions are relying upon needs to be highly up-to-date and comprehensive.

The paradigm shift from reactive to real-time and even pro-active decision-making is of key importance for the development of services that are designed to meet market needs. Incrementally building and deploying services that rely on deep customer insight and real-time feedback is expected to create services with a higher customer hit rate that can be developed more rapidly [2].

Rometty stated in [3] that "responding to change for gaining competitive advantage in the era of smart decisions will be based not on gut instinct, but on predictive analytics". The paradigm of "Open Innovation" emphasizes on the range of opportunities that are available to get access to distributed knowledge and information. Open innovation includes methods such as crowdsourcing that has been successfully used to develop new products. It has been proven as a beneficial approach for companies and users [4] as it integrates internal and external ideas and paths to market at the same level of importance [5]. It also makes use of distributed talent, knowledge and ideas in innovation processes [6]. We are proposing *Analytical Open Innovations (AOI)* to emphasize the role and importance of analytical methods in conjunction with the data and information retrieved from open innovation.

Service portfolio design consists of several decision points. We claim that a systematic process in combination with utilizing data analytics will facilitate better decisions. The *main contributions of this paper* are (i) proposing a systematic method called *AOI* which supports the different steps of the advance service portfolio planning (ASPP) process and (ii) performing a confirmative and exploratory case study mining the Android app market for Over the Top TV (OTT) services (i.e., video on demand, electronic program guide, etc.). The paper extends the results of a former conference publication [17] related to Value-Optimized Service Portfolio Planning. For the service evaluation, we now apply the Kano theory of product development and customer satisfaction. This is demonstrated to provide more comprehensive information on the acceptance and usefulness of the services. The case study is based on mining the Android app store and includes 36 atomic services extracted from 271 existing apps. Also, the process, content and scope of the crowdsourcing process have been enlarged.

The rest of the paper is organized as follows. In the next section, we will discuss background methodology in terms of quality driven service planning, open innovation, wicked service decision making and related information needs. In Section 3, we perform problem modeling, leading to the overall problem statement. The AOI solution approach is described in Section 4 and a detailed case study is described in Section 5. Finally, in Section 6, we provide a summary of our findings and give an outlook on future research.

2 Background and Related Work

2.1 Information Needs for Release and Service Portfolio Decisions

Different types of uncertainty make release decisions difficult. The individual and collective value of services is hard to predict as the value depends on the number of factors which themselves are dynamically changing (e.g., competition, market trends, user acceptance). One way to mitigate these risks is to perform a retrospective analysis on existing services based on semantic similarity; the result of the analysis can then be utilized to predict the value of the future services.

Historical attempts for creating products and services to answer the market's needs changed user's role and market focus. These attempts further continued by innovating products and services for higher market share and later transformed into innovating with customers to create real time value in the decision making process inside organizations. Aligned with this trend, open innovation helps product managers to achieve customer insight by using Web 2.0 functionality. As software projects grow and become more complex and by having more stakeholder across geographical and organizational boundaries, project managers increasingly rely on open discussion forums to elicit requirements [7]. This approach leads us to use open innovation as the rich source for providing data for the project.

The concept of a wicked planning problem was introduced by Rittel and Webber [8]. Planning and design problems are considered to belong into this category. Wickedness, refers to the difficulty to explicitly formulate the problem. To mitigate the challenge of wickedness, casual modeling is used to formalize judgmental process [9]. Release and service portfolio planning is considered as a semi-wicked problem and tackled with several approaches. Ngo-The and Ruhe [10] applied an evolutionary modelling and problem solving approach for mitigating wickedness [10]. A dialogue and explanation based approach was designed in [11] and a stakeholder-centric approach was proposed in [12] by Heikkilae et al. which emphasized on the need to broaden the scope of information input at the different stages of the whole planning process [12]. The former approach would be called closed innovation with respect to what we are proposing in this paper. Besides that, we are considering service portfolios, by focusing on the values they bring to the customer. Furthermore, we are able to utilize value and effort synergies as part of the final optimization process.

More specifically following information is needed for release decision making:

- Features or services: Wiegers [13] defined a product feature as a set of logically related requirements that provide a capability to the user and enable the satisfaction of business objectives. Also Lovelock in [1] defined services as economic activities focusing on time-based performance that brings desired results for recipients.

- Feature/service dependencies: Carlshamre et al. [14] reported that features are frequently dependent on each other. Based on their report 80% of features detected as dependent ones in the case of industrial software product release planning projects in the domain of telecommunications. Different types of dependencies such as implementation dependency, feature effort dependency, feature value dependency and feature usage dependency. Elicitation of dependencies as a type of knowledge elicitation is difficult. And ignoring the dependencies will create plans lacking to fulfill required conditions between features [15]. We also address the elicitation of service and feature dependencies by applying morphological analysis in [15,16].
- Feature/Service value: definition of value is context and user specific, and context changes will influence the value. Consequently, the value of a feature is time-dependent affected by changing market or business conditions. Furthermore, the individual value of a feature is not additive towards the overall release value and offering them in conjunction will provide synergies that are important to be taken into account [17]. A value-map is suggested by Khurum et al. [18] by using knowledge from state-of-the-art in software engineering, business, management, and economics, gathered through literature reviews and by interviewing industrial professionals in that the authors studied a broad range of value constructs and classified them as belonging to customer, internal business, financial, as well as innovation and learning perspectives.
- Stakeholders: Stakeholders play different roles in the definition of the product. Having different expertise in the context of product development their opinions are often contradictory. This causes a challenge in product release planning to determine a well-balanced product plan which addresses the most relevant concerns. Sharp et.al in [19] presented an approach on constructing the “the right set” of stakeholders. Within this, four groups of stakeholders are summarized:
  - Users and customers,
  - Developers,
  - Legislators (such as professional bodies, trade unions, legal representatives, safety executives or quality assurance auditors), and
  - Decision-makers (such as the CEO or shareholders).
- Stakeholder opinions and priorities: This process refers to the determination of priorities for the features/services under consideration for the next release(s). With the range of different segments, stakeholder opinions need to be clustered. The process of attracting stakeholder priorities is complex and reliable information is hard to get. However, without this information, product development becomes risky. Gorschek et al. found that critical stakeholders are overlooked and priorities are given in an ad hoc fashion [20]. More recently, discussion forums and user groups have been used to get access to stakeholder opinions [7]. Social network analysis is another direction to improve the requirements prioritization process. Fitsilis et al. [21] applied meta-networks where basic entities are combined for prioritizing requirements Targeting the customer value creation, Nayebi & Ruhe in [15,16] introduced crowdsourcing to provide a large set of potential customers and showed the increase of customer value within release plans.
- Resource consumptions and constraints: Development of features/services and later testing and integration of them into the existing product consumes effort and includes different type of human resources. Planning without investigating the resource constraints is not actionable. However, effort prediction is known to be very difficult and impacted by many factors. Different methods are applicable for providing estimates [22] Most of these methods incorporate some form of learning based on information available from former projects. In all the cases the up-to-date information related to factors influencing effort estimates are needed [23].

2.2 Open Innovation

Close provider and consumer relation in service firms puts a high premium on overall visibility and accountability [24]. Open innovation as a cheap and low risk problem solving approach relies on knowledge exchange with outside of company as a competitive advantage. Different forms of open innovation; crowd source, open source and outsource [6]; facilitate the provider and consumer interactions. Chanal [25] defines open innovation as a paradigm which assumes that firms can and should use both external and internal ideas and paths to market.

Rohrbeck et al. [26] presented an empirical study on incrementing the innovation capacity by changing the innovation processes and detecting some innovation enablers with discussion the comprehensiveness and effectiveness of them in their study. Later, Jansen et al. [27] presented an enterprise model to measure the software producers' degree of openness. Open software enterprise model [27] measures the degree of openness in a software producing organization.

From the business point of view, open innovation is the collective term describing business collaboration which combines internal and external ideas into architectures and systems whose requirements are defined by a business model. The business model utilizes both kinds of ideas to create value and define internal mechanisms to claim some proportion of that value [5]. In [28] involving customers in all phases of production was suggested along with using the agile process.

Crowdsourcing, open sourcing and outsourcing are types of open innovation [29]. Open sourcing is generally enabled by a web-based innovation platform [6] and will result in a product that is increasingly better, developed collectively and democratically [30]. Using Web-based forums to gather requirements from stakeholders is a common practice in open source and increasingly common across a number of enterprise-level projects which can grow quite large, with many thousands of stakeholders [31]. Open-source production is the exact spirit of peer production outsourcing a task or problem to a much wider pool of organizational and/or individual innovators.

Crowdsourcing as a type of open innovation is often enabled by the web [6] which is a creative mode of user interactivity [30]. Crowed wisdom in all of these approaches aggregates a collection of complementary data and information. Crowdsourcing is identified with other web 2.0 technologies [32]. A large chunk of the Web is about data and services. Consequently, we expect crowdsourcing to build structured databases and structured services (web services with formalized input and output receive increasing attention these days.) [33]. The idea of using a large pool of problem solvers has been discussed in different contexts [6]. Further, instead of simply collaborating with a selected set of known external parties, firms are innovating by using crowdsourcing [29] Crowdsourcing is a participative distributed online process that allows the undertaking of a task for the resolution of a problem and any non-trivial problem can benefit from crowdsourcing [34].

Currently, several open innovation platforms and services are available in different scopes [35] [36] but none of them are specialized in the domain of software service/product lifecycle. In [37] the need of internet based software was analyzed for enabling knowledge management along with the processes and internet based platform for managing innovation at the business process levels were introduced. Paradigm of Open Innovation is targeting the usage of better knowledge and information to qualify products decision-making. Information needs in software development and management was studied by Buse and Zimmermann [38].

Wnuk and Runeson [39] present an open innovation framework for software engineering where they point into the need of software processes that supports open innovation. West et al. [40] conducted a review on open innovation and integrated the process introduced among all the studies into the following three steps:

- Acquiring innovation from external sources – this step includes the search for innovation and later enabling and contracting the innovation.
- Integrating innovation – which not only facilitates the integration but also defines the activities that are changing in an innovative way.
- Commercializing innovations

Achieving new markets with better cost effectiveness in production leads firms to collaborate with each other in the form of outsourcing. In this case Pre-selected firms provide goods or services to the client with a clear compensation process. In this study we adopt the open innovation with analytical techniques to know the customer values and the current state of the market.

2.3 Kano Model for Customer Driven Portfolio Planning

As the services to be offered are intangible and determining the demand for them is difficult, understanding service production and consumption is complex. This process needs to be attuned to the needs of consumers to meet their demands successfully. A key part of this process is responding to existing and emerging consumer trends even if they are conflicting. The traditional view of release planning favors the delivery of pieces of functionality in a best possible way. Resource consumption should be considered for a feature or service planning in a product release and the consumption of resources in a way to be returned via the product delivery to customers is of importance.

From this point of view, finding a plan to bring customer value by fulfilling their satisfaction is of importance. Consequently, the problem now becomes a trade-off analysis where the balance is between providing the most comprehensive and satisfactory functionality in dependence of the returned value by these services. A tool for supporting release planning of quality requirements and its initial industrial evaluation has been presented in [41]. As part of the underlying planning method called QUPER [42], the tool helps to reach an alignment of what level of quality is actually needed by different groups of stakeholders.

The Kano model is primarily proposed to analyze that which "products and services can be used to obtain a high level of customer satisfaction". This model replaced the one-dimensional relation between perceived product quality with the satisfaction level of customers with the impact analysis of each feature and service on customer satisfaction [43]. The very first Kano model was introduced in the early 1980's [44] and since then it has evolved and been integrated into other models of design and requirements engineering such as Quality Function Deployment [45]. Mikulic [46] classifies Kano technique as highly reliable for classification of quality attributes at the design stage. The in depth analysis of responses in Kano questionnaire helps with customer segmentation [43].

The following deficits triggered research for enhancing the traditional Kano model:

- The lack of quantitative analysis assessment making it impossible to differentiate the significance of assigning each service to a Kano requirement category [47].
- The strict classification in traditional Kano model limits decision making support [47].

The main problem is precisely in defining fulfillment and non-fulfillment of customer satisfaction. This may be caused by problem in question statement. As discussed in [46] the questions should explicitly target the provision/non-provision of a quality attribute and faulty language may transform the question about the performance of the quality attribute instead. In the case that the questions have not been properly formed, the Kano model may provide mixed results and loses its accuracy [46].

The literature review by Luor et al. [37] reported 94 articles on Kano between 1998 and 2012. They reported 22 major articles on this model which most of them where using the traditional Kano for empirical studies. Chen et al. [48] proposed an integrated Kano process to optimize quality considering several criteria and analyzed that the usage of Kano helps to differentiate between the factors affecting customers' satisfaction. They examined their proposed approach within a case study.

Lee et al. [49] proposed fuzzy Kano questionnaire to cover the deficit of traditional Kano questionnaire that pointed above and made it less subjective. They tried to extent the traditional approach by providing a questionnaire to help customers extend their feelings utilizing membership and assigning numeric degrees to their wills. Later Xu et al. [47] proposed the analytical Kano model to evaluate the customer satisfaction levels in conjunction with organizational capacity. They proposed Kano classifiers as tangible criteria for categorizing customer needs. They also introduced a decision factor for supporting decisions of configurations design. Their analytical approach provides a quantitative measure of customer satisfaction based on Kano.

3 Modeling and Problem Statement

Service release planning is targeting the delivery of optimized portfolios of the services which are most valuable to subscribers. Here, we are providing a formal model and a subsequent analytical approach. As any model, it is impossible to accommodate all details. However, adapting the principles of both open and closed innovation is intended to increase the validity of the model and the data used in it. In what follows we introduce the necessary concepts and notation to provide the formal problem statement.

3.1 Services and their Dependencies

The key unit of investigation in this paper is services and their composition into service portfolios. We assume an existing set called SERVICES consisting of all given candidate services. Later on, in our case study, we consider apps being the portfolio of services. We call these services as *service(i)* (i = 1 ...n). For a given time horizon (and related number of iterations), not all services can be offered. Selection of the most attractive ones and their combination to achieve highest overall value is the key content of service portfolio planning. In the case of apps, the portfolios are a collection of apps covering different ranges of functionality.

Services depend on each other for different reasons. This might be related to their usage (one service requires another one to be meaningful for the user) or in their actual implementation. First, the set of coupling dependencies are presented called *CSD*. CSD is the set of mutually coupled services requested to be offered in the same iteration. A formal definition is given in (1):

$$x(i) = x(j) \text{ for all pairs of coupled services } (service(i), service(j)) \in CSD \tag{1}$$

Similarly, the set, called *PSD,* of precedence dependencies between services is defined by:

$$x(i) \leq x(j) \text{ for all pairs of precedence related services } (service(i), service(j)) \in PSD \tag{2}$$

Finally, NAND dependency is defined. As described in [16], certain services are not compatible with each other and do not make sense to be offered in conjunction. Detection of these incompatibilities is a complex problem itself. In [16], we have used morphological analysis in combination with Cross Consistency Assessment (CCA) to solve this problem. NAND indicates that:

$$x(i) \text{ NAND } x(j) \text{ if and only the two services } (service(i), service(j)) \in NAND \text{ cannot be offered in conjunction} \tag{3}$$

3.2 Effort of Services

Each $service(i)$ requires an estimated amount of effort called *effort(i)*. Effort estimation is known to be inherently difficult. Not being the focus of this paper, we assume that effort estimation is in place and that one of the variants of estimation by analogy was applied to provide the data.

In the problem formulation presented here, we assumes capacities called $capacity\ (k)$ for the different time intervals (e.g., quarters of a year) are considered for planning. For simplicity, we do not differentiate between different types of resources and look into effort subsuming all relevant types of effort needed to provide the service. In total, this results in K constraints of the form described in (3):

$$\sum_{i:\, x(i)=k} effort(i).x(i) \leq capacity(k) \tag{4}$$

3.3 Crowdsourced Kano Evaluation of Services

Each service has an associated value. The value is determined from customer and user evaluation of all the services. Value projection is difficult and uncertain in its nature. For the AOI approach described in Section 4, we make two key assumptions to facilitate this process. The first one is to utilize crowdsourcing with a larger number of responses to determine the likely value of services. The second one is replacing single point value evaluation by the more comprehensive Kano approach [50] evaluation.

Based on Kano model [50] the questionnaire consists of two questions per service. The first question (called *functional*) evaluates the customer's reaction in presence of a specific service. The second question (called *dysfunctional*) evaluates the customer's reaction in the absence of the same service. The answer of each question is designed in a five point Likert scale defined as below:

- ✓ I like it that way
- ✓ It must be that way
- ✓ I am neutral
- ✓ I can live with it that way
- ✓ I dislike it that way

By combining the two answers of functional and dysfunctional questions (see Table 1), each service can be classified as one of the following categories:

- Must-be (M): The prerequisite requirements that customer assumed them as granted ones and availability of these requirements is leading into the “not dissatisfied” state but it does not affect the satisfaction level.
- One-dimensional (O): The fulfillment of these requirements will linearly increment the level of customer satisfaction.
- Attractive (A): These requirements are not explicitly requested and the absence of them won’t cause dissatisfaction although availability of them is expected to lead into great customer satisfaction.
- Indifferent (I): The customer feels indifferent towards the availability of this service.
- Reverse (R): Stated that the customer not only does not want this service but also expected the reverse.
- Questionable (Q): This category is the indicator of a problem in phrasing the question or understanding the question which resulted in crossed out answers.

Table 1. Kano evaluation table [50].

Customer Requirements		Dysfunctional questions				
		Like	Must-be	Neutral	Live with	Dislike
Functional questions	Like	Q	A	A	A	O
	Must-be	R	I	I	I	M
	Neutral	R	I	I	I	M
	Live with	R	I	I	I	M
	Dislike	R	R	R	R	Q

For each service, integrating the above evaluation process to the crowdsourcing process, defined the number of services in each Kano category. We eliminate the category Q as it indicates a problem in either phrasing or understanding the question. Assuming that N individuals answer the Kano questionnaire, for each service *service(i)* we will have a five-dimensional vector called *KF(i)*. The components of the vector correspond to the different types of Kano classifications (M, O, A, I, R). The actual values of the components are called $KF_X(i)$ where $KF_X(i)$ gives the number of services in category X. Consequently, we have

$$\sum_X KF_X(i) = N \text{ for all services } i \tag{5}$$

Based on the Kano evaluation performed by the crowd, we can define the value proposition of an individual service. This is defined as the linear combination of the frequencies with an assigned factor of importance for each Kano category. The underlying rationale is that the decision maker can determine the relative importance of offered services based on environmental factors based on software product strategies (for example having M services compared to O type of services). The relative weights are called *weight(X)* with X representing the category again. Weight is defined on a ten-point scale {0,1, ..., 9} with the interpretation that the higher the value, the higher the relative importance of the category.

Based on all the above, we define the crowdsourced Kano value of an individual service by (6):

$$KFvalue(i) = \sum_X weight(X)\, KF_X(i) \tag{6}$$

3.4 Value of Service Portfolios

The value of a service portfolio is the sum of the discounted services offered within the portfolio. The release of service portfolios is assumed to be done iteratively. With K iterations (also called planning horizon), each iteration has a discounting factor *discount(k)* reflecting the decreasing value of later delivery of services. In general, and just for conformance to the tool user later on (other scales and measures would not change the method), *discount (k)* is taken from the interval {1 ...9} with

$$discount(k) \geq discount(k+1) \text{ for all } k = 1 \ldots K-1 \tag{7}$$

As a consequence of (7), services offered at an earlier stage will contribute more to the overall utility of the service portfolio. A service portfolio called *servport(n)* is a set of atomic services offered over the given period of time of K iterations. Applying all the notation introduced above, the value of a given service portfolio *servport(n)* is now defined by

$$\text{spvalue}(n) = \sum_{X} \text{weight}(X) \left(\sum_{k=1..K} \text{discount}(k) \left(\sum_{i:\ x(i)=k\ \&\ \text{service}(i)\ \text{from servport}(n)} \text{KFvalue}(i)\right)\right) \quad (8)$$

In summary, the value of a service portfolio combines the perceived value contributions gained from all categories, taken over all the services offered in this portfolio over the K time intervals.

3.5 Problem Statement

Now, a formal statement of the problem under investigation could be formulated. We call it Advance Service Portfolio Planning (ASPP). ASPP is based on the linearity assumption of the portfolio effort. More precisely, the total effort called *effort(n)* consumed by a service portfolio servport(n) is defined as the cumulative effort consumed across all iterations for all the services offered in this portfolio.

Problem ASPP: For a given set of candidate services SERVICE, determine all trade-off solutions SERVPORT that are

(i) feasible in terms of cost and technology constraints and are

(ii) non-dominated by any other service portfolio in terms of their added value to the effort consumed.

$$\begin{aligned}&\mathit{SERVPORT = max\text{-}min\{Trade\text{-}off(spvalue(n),\ effort(n))}\\ &\quad \mathit{among\ all\ portfolios\ created\ from\ SERVICE)}\end{aligned} \quad (9)$$

In ASPP, the search is for trade-off service portfolios balancing the goal of maximizing the value of the portfolio while minimizing the effort needed to create it. Non-dominated (also called Pareto) solutions here means to have a portfolio which cannot be improved in one dimension without compromising (getting worse) in the other one.

4 Service Portfolio Planning Approach AOI

The overall AOI architecture (as illustrated in Figure 1 is designed with the aim of (i) involving users in the process of innovation and in the development of products along with stakeholders as the traditional source of service decisions, and (ii) then tailoring customer needs towards the organizational constraints by considering effort needed to implement services as well as considering product strategies for planning.

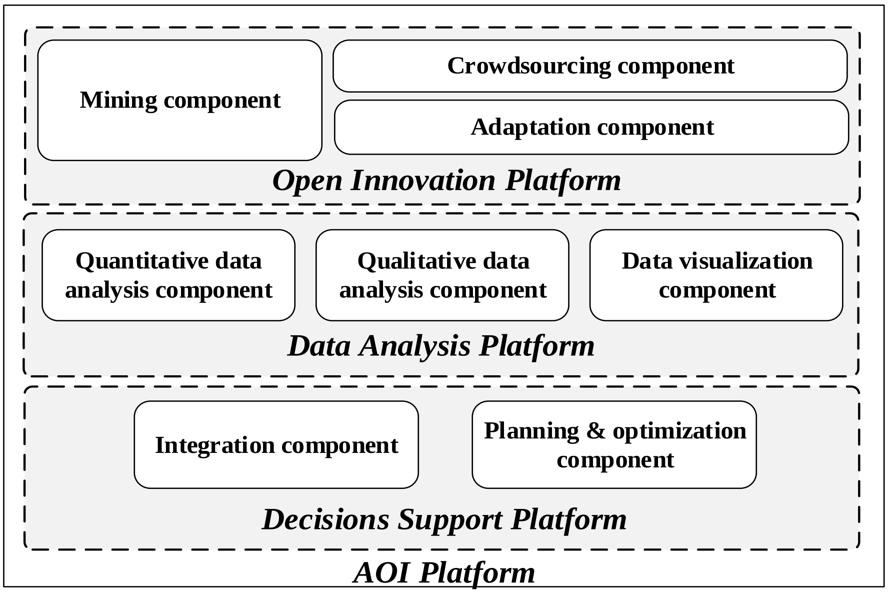

Figure 1. Proposed AOI architecture.

The proposed AOI approach consists of three main platforms that are described below:

Open innovation platform

1. *Mining component:* In order to gather data from web and to gain knowledge on the context, we utilize crawler software which incrementally mine and update the data from destination websites (i.e the app markets). We parse the data and forming structured database and extracting the needed data (such as app information).

2. *Crowdsourcing component*: With the aim of user involvement, crowdsourcing is used. The crowdsourcing platform provides the crowd for answering questions and for facilitating control and verifying their works and task distribution. By now, Amazon Mechanical Turk [51] service, as a micro-task market, is employed which help us to get access to large pool of potential stakeholders. At the time questionnaire is designed based on Kano approach for gathering the expectations and satisfaction of potential customers.

3. *Adaptation Component:* The collaboration with experts was maintained through a tool named Very Best Choice™. This software is used to manage collaboration between systems and to control the representation of feedback and to provide in house collaboration and within organizations. VBC light is a lightweight decision support system designed to facilitate proper priority decisions [52]. Moreover, text mining platform is used along with other platforms to enable automatic understanding of the crowd's response to generate meaningful data. Other than effort estimation, the experts can discuss the organization strategies, vote on them and choose the product strategy (like the weight vector introduced in Section 3.5) and define the factors for portfolio planning.

Data Analysis Platform

Moreover, quantitative data analysis platform is used along with other platforms to enable automatic understanding of the crowd's response to generate meaningful data.

1. *Quantitative data analysis*: Analyzing, structuring and extracting services from descriptions (i.e., the services from app descriptions) needs a powerful text mining and topic modeling tool. Several commercial tools are available for this purpose. "Content analysis" [53] is the methodology used for extracting and structuring data from the information gathered. Using a commercial tool[1] for this type of analysis, the text is decomposed into categories with relations among them and several inferences are provided. To extract services from app descriptions, we conducted a conceptual analysis on top of the content analysis methodology with the commercial tool.

2. *Qualitative data analysis*: This component consists of several analytical techniques leading to solving the problem to get the inference from the data this platform consists of modules which are adapted with three technology pillars employed in a successful analytics platform [54]. This layer is interpreting and meaningfully structuring the data. Working on value-effort analysis, we effort calculation and value calculation are part of this platform.

Decision support platform

1. *Integration component*: Considering the inter-organizational needs for planning portfolios practically needs the consideration of product strategies along with capacities and their returned value. Considering the data retrieved from the context (i.e. app stores), the outperformed portfolios are extracted and the data passes for comparison with our optimized portfolios.

2. *Planning component*: For planning, the tool named ReleasePlanner™ was used. This pivotal platform provides a proven [55,12,56] functionality to facilitate prioritization as well as generating optimized portfolio plans. The optimization module is responsible for the computation of optimized and diversified alternative release plans based on specialized integer programming and the special structure of the problem. The analysis and decision module defines alternative services and plans with their resource consumption and degree of stakeholder excitement.

[1] http://www.leximancer.com/

5 Case Study

5.1 Overview

We presented a case study on the OTT services in our predecessor paper [16]. Here, we will present a more comprehensive and advance case study on Over the Top TV (OTT) services [57] [58]. In its nature, our case study is a confirmative one to confirm our proposed approach. Therein, as a proof of concept, we show the usage of the proposed AOI approach in designing a new app, thereby utilizing all the knowledge gained from open innovation. In the terminology introduced in Section 3, we look at apps being service portfolios with feature apps being the atomic services.

As a first step of the case study, we extracted services that are suitable for providing a comprehensive portfolio of services for OTT. We did this by analyzing the services of available and related apps in the Android app market. Later, we planned for a portfolio of these services with the objective to maximize customer value for the app publisher. We performed crowdsourcing to align extracted services with customers' desires. We performed crowdsourcing on the services extracted from app descriptions as a means to predict the degree and type of acceptance of the services.

In this case study, we illustrate the effect and process of applying the AOI approach in practice. In particular, we demonstrate the process of gathering data from app stores, generating data from crowdsourcing and making decisions based on context specific needs. In addition, we analyze the effects of applying AOI on planning for an app that provides better customer satisfaction by using the Kano evaluation. After all, based on the market data and crowdsourcing data, we generate service portfolio plans and evaluate levels of customers' satisfaction. Besides showing the applicability of our approach, we show that the proposed apps (service portfolios) are outperforming the existing app offers in terms of their value to cost ratio.

The process of the case study is described in Figure 2 based on the proposed AOI platform explained in Section 4.

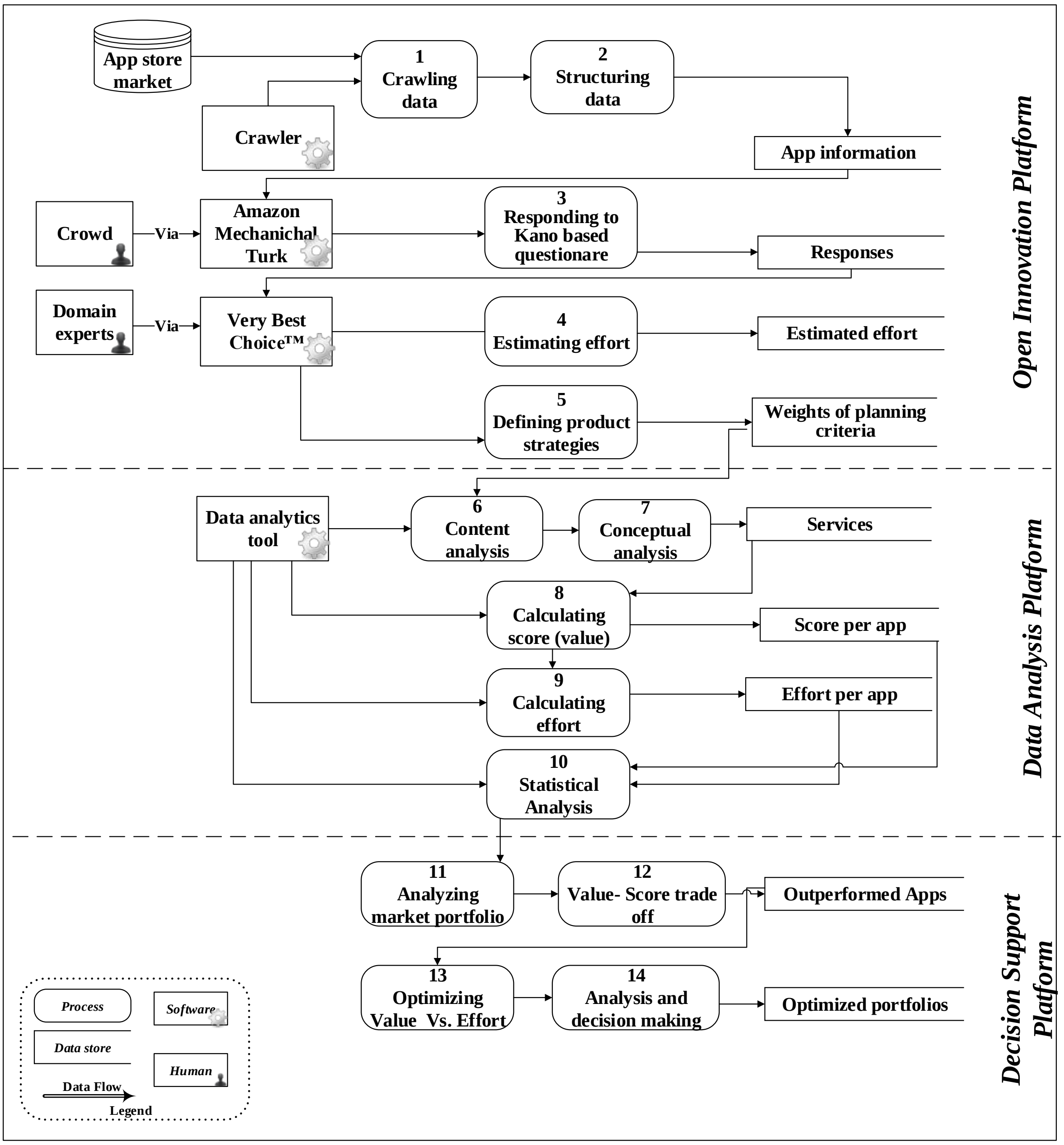

Figure 2. Case study process steps and their mutual dependencies.

5.2 Methodology for Data Gathering

Our case study methodology involved three types of data gathering being (i) mining the Android app market (ii) crowdsourcing and (iii) expert based estimation of effort for OTT service development. In what follows, we describe the main content and results of these processes. Further details can be found on our website[2].

5.2.1 Android App Store Mining

Protocol. The case study is focused on Android app stores via Google Play. App stores in general provide a unique source of data by collecting the customers' feedback in the form of comments, ratings and numbers of

[2] http://ucalgary.ca/mnayebi/data-sets

downloads. Harman et al. in [59] utilized this data to extract services and analyze related information. To triangulate data [60], we gathered apps from two different sources, being (i) the official domain of GooglePlay and (ii) the Appbrain[3] website (Googleplay third party analytical system). We compared the 60 apps retrieved from Appbrain with the data gathered from GooglePlay. We observed that Appbrain is data intensive (i.e., it includes the versioning history) but no differences were observed in the apps' descriptions. As a consequence, we based our analysis on the GooglePlay samples and data. This data gathering phase is reflected in Process 1 and Process 2 of Figure 2.

Subject. We gathered the third degree data [60] by using some web crawler software[4]. The data of over 7000 apps where gathered, and among these apps, three keywords- "OTT", "IPTV" and "Internet TV"- were searched that retrieved 271 related apps. We sought out services by targeting the parts tagged as "Description" and "What's new" among the crawled information.

Data Analysis. Our focused data is qualitative, though we used "content analysis" [53] as the methodology for extracting services from the description of apps. Using a commercial tool for this type of analysis, the text is decomposed into categories with relations among them and several inferences are provided. To extract services from app descriptions, we conducted a conceptual analysis on top of the content analysis methodology with the commercial tool. By this, we extracted the compound concepts from app descriptions as services.

5.2.2 Crowdsourcing

Protocol. In order to extract customer satisfaction values a questionnaire (see Appendix I) was designed based on the Kano model. For each of the apps services extracted in Section 5.2.1, we asked questions to measure both satisfaction and dissatisfaction levels. Crowdsourcing was done through an online survey. In order to extract level of satisfaction/dissatisfaction about the TV services, our questionnaire was designed with a five point Likert scale.

The submitted task was published in Amazon Mechanical Turk®[5] with the capacity of 125 respondents and response duration of seven days. The submitted tasks were completed in five days and among 125 answers, 100 answers were qualified. The qualification was done based on the following criteria:

- Time allotted to answer questions- The questionnaire had 72 questions and we estimated that it would need about 30 minutes to be completed. The responses which took significantly less than this duration, were excluded. In total, four of the responses were excluded because of this reason.
- Syntactical analysis of answers – The responses which were chosen same answer for several tandem questions were excluded. Applying this rule, we exclude 21 responses.

Each submitted task was awarded with a 0.25$ incentive in addition to a bonus for high quality responses. The bonus of 0.3$ were granted to 100 individuals. This data gathering phase is reflected in Process 3 within Figure 2.

Subjects. The workers were exclusively selected from North America. We requested for TV users who are interested in mobile apps to answer this task. As the majority of people are familiar with TV, this was not a concern in the context of this study. Gathering the personal information is restricted in Amazon Mechanical Turk. Consequently, no further information is available to characterize the population.

Data Analysis. Having users' responses, we define the frequency of responses in each Kano category – must-be, one-dimensional, reverse, attractive, indifferent and questionable. Furthermore, we utilized this result to define the value of each service.

5.2.3 Expert-based effort estimation

Protocol. In order to estimate the effort needed to develop a specific app, we gathered the data from contacting expertise as reaching the publisher of the apps was not possible. The estimations were done as a three-point estimate including worst case, expected estimation and best case estimation. This step is shown in Process 4 in Figure 2.

---

[3] http://www.appbrain.com/
[4] http://nutch.apache.org/
[5] www.mturk.com

Subjects. As reaching the publisher of the apps was not possible, three independent experts with experience in developing apps for OTT in both the Android and iTunes market were recruited. For the domain of OTT services, they performed independent estimation of the effort needed for developing each of the selected 36 services. The data gathering was supported by the VeryBestChoice[6] tool which facilitates the aggregation, estimation and later analysis of the data.

Data analysis. We utilized expert judgment and applied the three-point PERT algorithm. Having the worst, best and expected estimations per effort from the three experts, we calculated a weighted average based on the expertise of the subjects, applying weighting factors 9, 7, and 5, respectively.

5.3 Results

5.3.1 Android app store mining

We utilized a proprietary text analytics tool for analyzing descriptions of apps which we already retrieved from Android market. We extracted 36 services. The services and their frequency of occurrence is shown in Figure 3. As it may be observed, *Search* functionality is the most frequent service, followed by *Variety of quality support (HD, Ultra HD, ...)*. This is reflected in Process 6 and Process 7 of Figure 2.

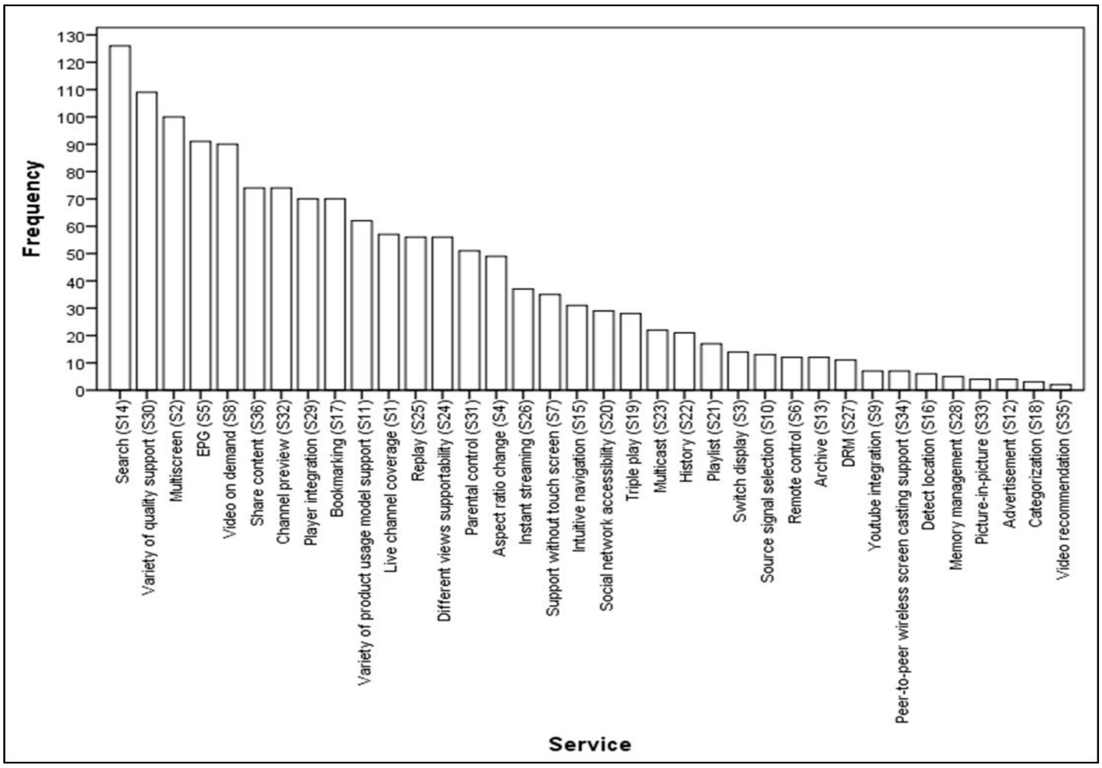

Figure 3. The occurrence of services across the 241 selected apps.

Analyzing all the apps in our sample, we retrieved 36 services; in which the most comprehensive app included 22 services. We also eliminated the apps which were not disclosing any services. 30 apps did not have representative description and excluding them resulted in the sample size of 241.

[6] http://edi.lite.verybestchoice.com/

5.3.2 Crowdsourcing

From performing crowdsourcing, we extracted the frequency of occurrence of the different Kano dimensions for each service. The results of this analysis are shown in Figure 4. To do so, we followed the Kano approach and mappings described in Section 3.3.

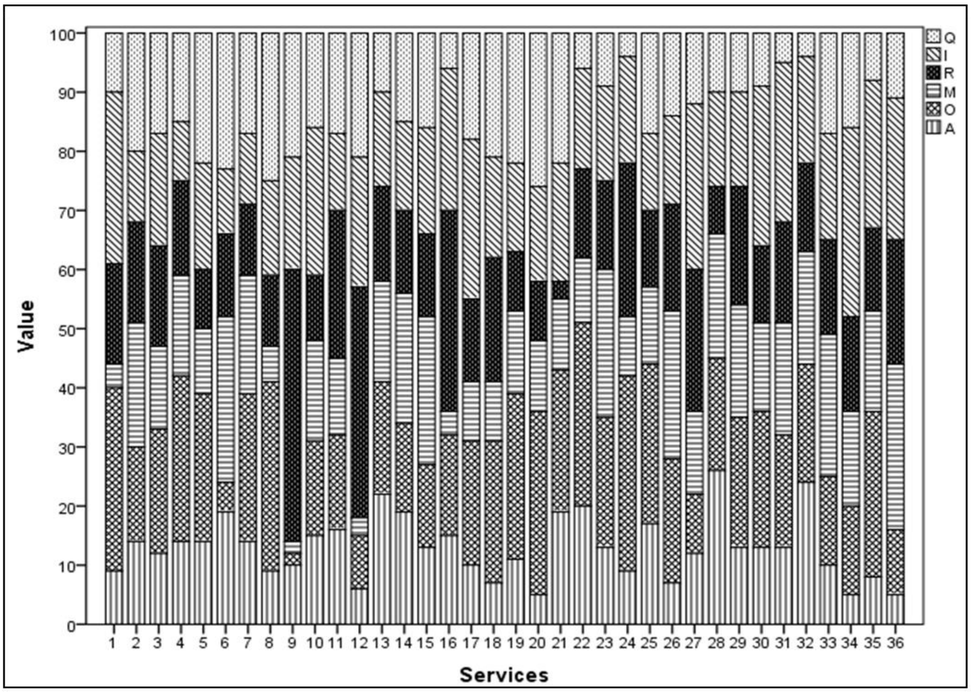

Figure 4. Frequency of occurrence (across 241 apps) per Kano dimension (Questionable (Q) /Indifferent (I)/Reverse (R)/Must-be (M)/One-dimensional(O)/Attractive (A)) for all crowdsourced services.

The questionable (Q) scores indicate that the result is not clear and the person answering the question has likely misunderstood the question or chose the wrong answer. The Q related data is not used for creating optimized portfolios in Section 5.4. As is shown in Figure 4, each service has some portion of each Kano category (the descriptive statistics are provided in Table 2). Following the Kano model, each service was assigned to the category for which the crowd voted most frequently, as discussed in Section 2.3. For example, as it can be seen from Figure 4, Service 1 has the highest portion of frequency in the "One-dimensional" category, and we have it classified as such.

Table 2. Statistics of services related to each Kano category.

Category	Min	Max	Average	Variance
Must-be	0	6	1.93	1.35
one-dimensional	0	11	3.17	5.21
Indifferent	0	5	1.37	1.36
Reverse	0	2	0.57	0.32
Attractive	0	2	0.29	0.23

5.3.3 Expert-based effort estimation

We calculated the weighted average of the effort needed for developing each service. Figure 5 shows the effort we considered for developing each service. The detailed data is provided in the Appendix II.

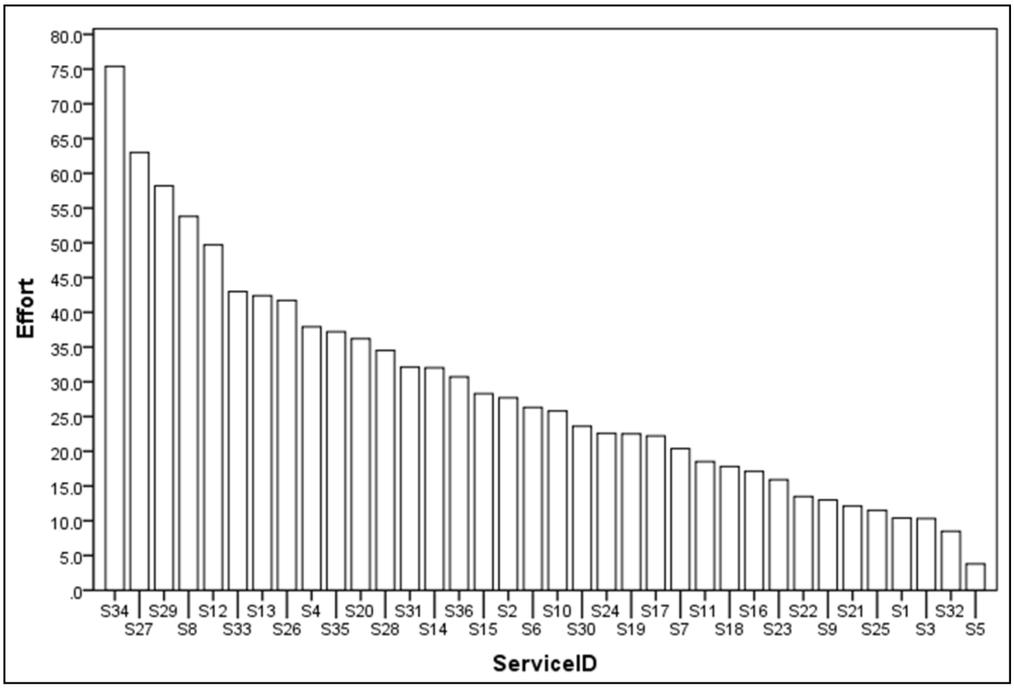

Figure 5. Services in decreasing order of effort estimate (in person days).

5.4 Service portfolio optimization

The final objective of our approach is the creation of value optimized service portfolios in the app market. With all the necessary information available, we are now searching for the solutions of our ASPP problem being optimized apps for the different levels of effort invested. More precisely we optimize a weighted value of the Kano evaluated and categorized services.

One of the key benefits of the Kano driven service planning approach is to allow putting varying emphasis on the different service categories. Indeed, the decision whether M (must-be) services are more important than A (attractive) services, and if so by how much, is context specific. Utilizing the terminology introduced in Section 3.3 and 3.4, we assign weights to the different service categories. With categories (M, O, A, I, R, Q), the weight vector is (8, 6, 4, 0, 0, 0). This means, the main emphasis is on M and then on A categorized services. The definition of these weights are derived by product strategies as reflected in Process 5 of Figure 2 and shows the emphasis of product developer to provide different service portfolios for various market segments.

Applying (6) with the Kano frequencies $KF_X(i)$ shown in Figure 4 (also see Process 8 in Figure 2) and calculating the effort needed to implement the services (Process 9 in Figure 2), we ranked the services by the value-effort ratio as it is shown in Figure 6. As a conclusion, the services with the highest ratio between value and effort consumed appear to be the most promising ones for the creation of the optimized service portfolios naming S5, S32 and S25 as the top 3 ranked services in Figure 6 This step corresponds to Process 10 of Figure 2.

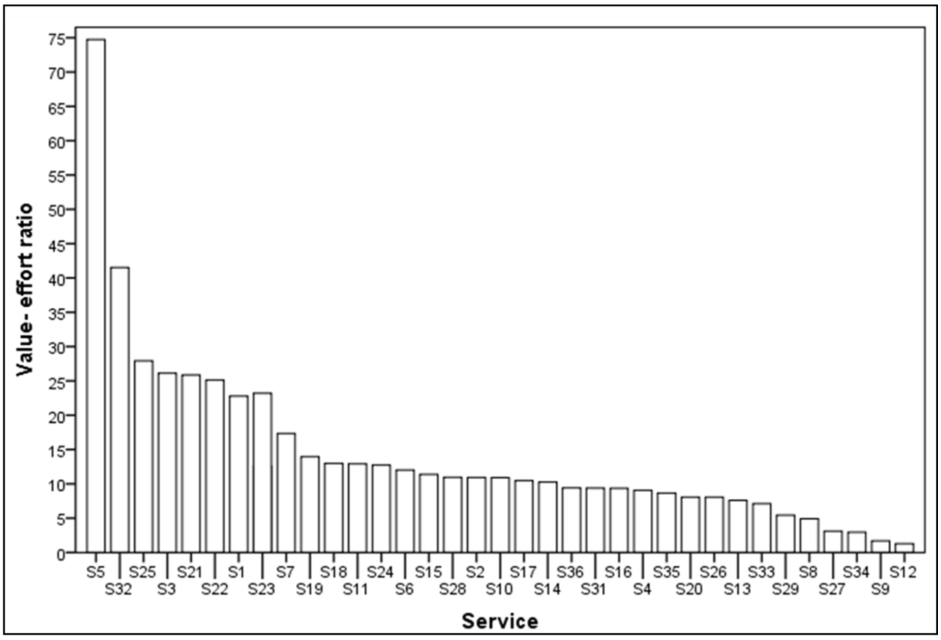

Figure 6. Ordered value per effort ratio across all services.

To demonstrate the improvement of the optimized service portfolios, we performed service optimization on different levels of total effort consumed. Aligning with the terminology of Section 3, and to keep the planning reasonable, we were planning for just K = 1 time period. As a result, we had to solve a series of knapsack problems [61], one for each level of effort.

For evaluating the 241 existing (non-optimized) service portfolios (apps), we performed a trade-off analysis which is reflected in Process 11 and Process 12 in Figure 2. Applying the service portfolio value computation formulae (8) for the case of K = 1 iteration results in a simplified function (10):

$$spvalue(n) = \sum_{X} weight(X) \sum_{service(i)\ from\ servport(n)} KFvalue(i) \quad (10)$$

Using the above agreed upon weight vector (8, 6, 4, 0, 0, 0) and also applying the atomic service effort estimates, results in a scatter plot of two-dimensional value-effort evaluation of all 241 service portfolios under investigation (see Figure 7). The actual data can be seen on the left part of Figure 8 with the boundary (dotted line) describing all the non-dominated (among the 241 alternatives) service portfolios. In total, 15 of these portfolios were found (showed in bold circles in Figure 7connected by the dotted line).

We demonstrate that our optimized service portfolio planning approach offered in this paper is able to generate better portfolios (Process 13 and Process 14 in Figure 2). For that, we fixed the different (15) levels of effort consumption and the generating optimized portfolio. From Figure 8, we can see that all these optimized solutions (showed in red squares along the solid line) are better than the best ones from the original service portfolio set at this level. The bigger the distance, the higher the improvement the optimized planning achieved. As can be seen from Table 3, the relative improvement (ratio of optimized versus non-optimized service) at a fixed effort level varies between 16.5% and 95.3%.

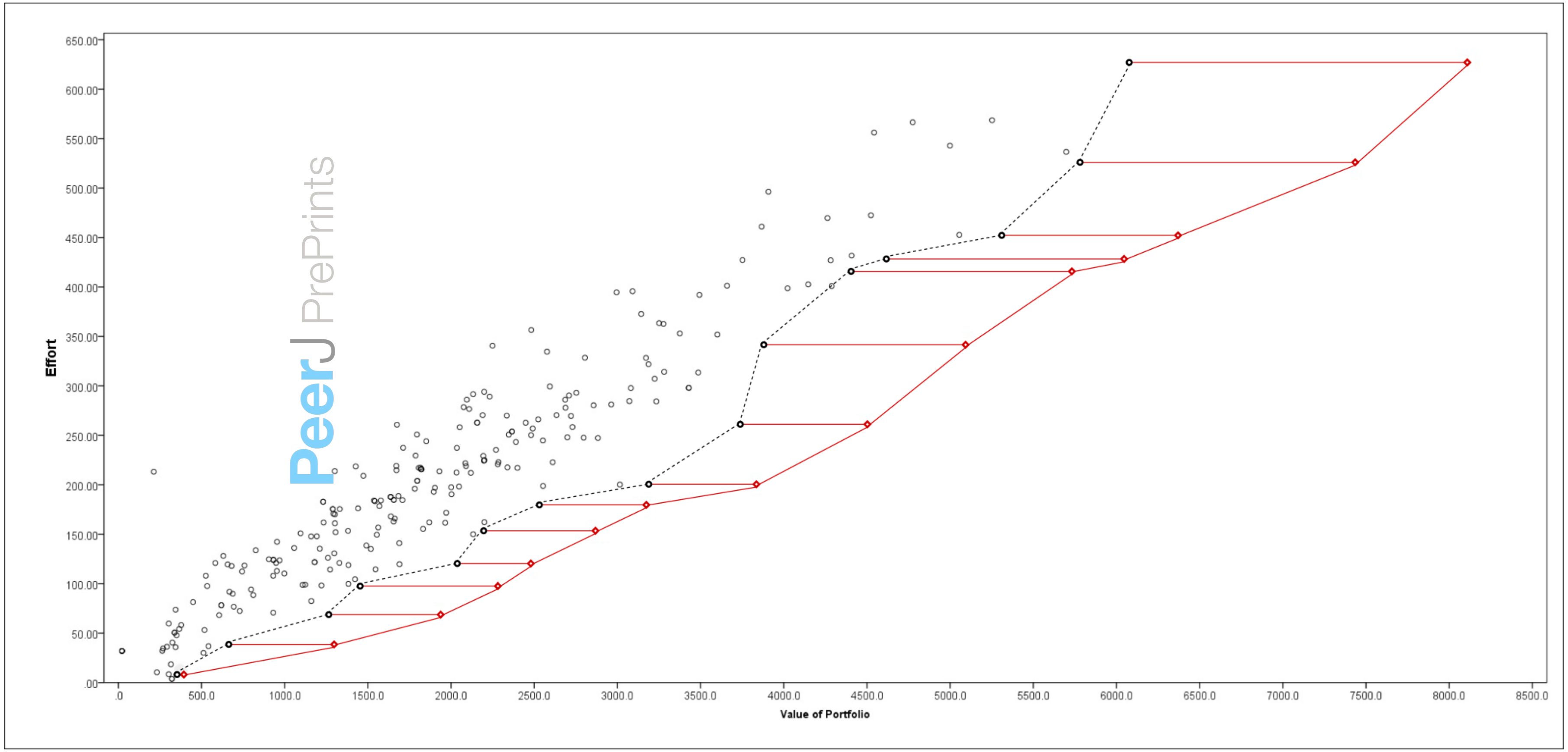

Figure 7. Comparison between actual service portfolios and their Pareto solutions at the dotted line versus optimized service portfolios (along the solid line).

Table 3. Value-effort optimized portfolios.

SERVPORT	Effort	Value	Value Increment in proposed Pareto plans (in %)
{S1,S2,S3,S4,S5,S6,S7, S10,S11,S13,S14,S15, S19,S20,S21,S22,S23,S24,S25,S26, S28,S30,S31,S32, S35, S36}	626.5	8111	33.4
{S1,S2,S3,S4,S5,S6,S7, S10,S11, S14,S15, S16, S19,S21, S22,S23,S24,S25,S26, S28,S31,S32, S35, S36}	525.6	7432	28.56
{S1,S2,S3,S4,S5,S6,S7, S10,S14,S15,S19,S21, S22,S23,S25,S26, S28,S32, S35, S36}	452	6372	20
{S1,S2,S3,S4,S5,S6,S7,S9,S11,S16,S18,S21,S22,S23,S24,S25,S26,S28,S31,S32, S36}	427.9	6046	23.66
{S1,S2,S3,S4,S5,S6,S7,S9,S11,S16,S18,S22,S23,S24,S25,S26,S28,S31, S32,S36}	415.8	5733	30.06
{S1, S2, S3, S4, S5, S6, S7, S9, S11, S16, S17, S18, S21, S22, S23, S24, S25, S32, S36}	341.5	5263	35.61
{S1,S2,S3,S4,S5,S6, S7,S14, S19,S21, S22,S23,S25, S28,S32}	261	4506	20.55
{S2,S3,S5,S7, S11,S19,S21, S22,S23, S25,S28,S32}	200.2	3836	27.23
{S1,S3,S5,S11,S16,S21,S22,S23,S24,S25, S32,S36}	179.4	3461	36.64
{S1,S3,S5,S9,S11,S16,S22,S23,S24,S25, S32}	150	2957	38.63
{S3,S5,S9,S11,S22,S23,S24,S25, S32}	119.9	2484	17.87
{S3,S5,S7, S21,S22, S23,S25, S32}	97.6	2281	79.08
{S3,S4,S22,S23,S25, S32}	68.8	1935	52.6
{S5,S22,S25, S32}	38.5	1297	95.33
{S32}	8.5	353	16.5

5.5 Optimized and diversified service portfolios

As an additional analysis and enhancing the results provide in Section 5.4, we looked at the possibility to offer up to five alternative service portfolios for each level of effort consumption. These solutions are shown in Figure 9. All the alternative plans are near optimum, but designed to be structurally different (in terms of which services are actually offered). This is an application of the diversification principle [10] stating that "*A single solution of the cognitively complex problem is less likely to reflect the real-world problem solving needs when compared to a portfolio of qualified solutions which are structurally diversified. This is not changed by the formal optimality of the single solution*".

We have determined qualified and diversified service portfolios. In most cases (if diversification was not substantially compromising quality), five solution were found at each level. These solutions are presented in Figure 11. In a more detailed view given in Table 7, we provide additional information on the number of must-be, one-dimensional and attractive services per optimized portfolio. Overall, this information is offered to the app vendor to allow selecting a final portfolio that is in best match with the business and market context.

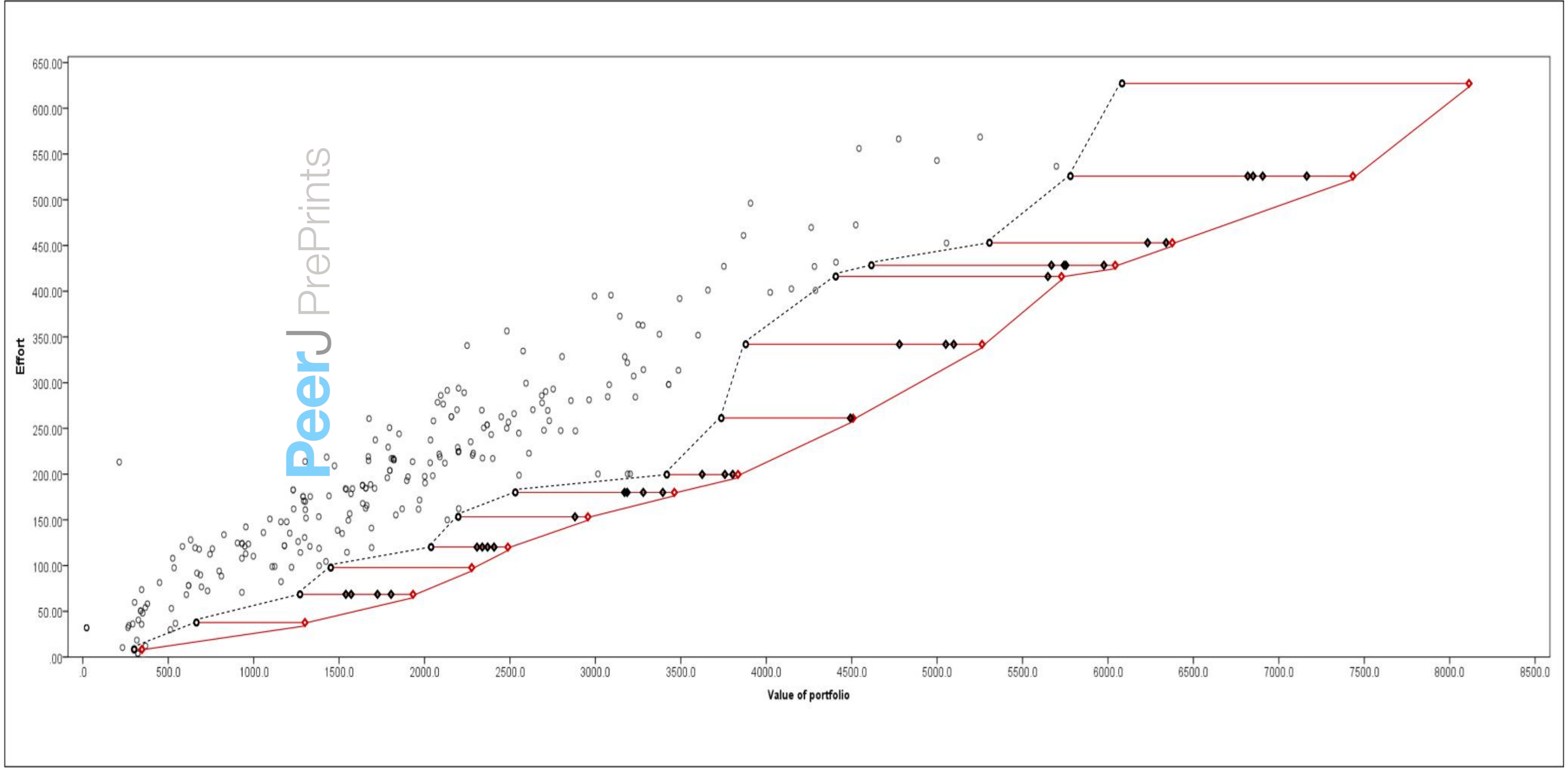

Figure 8. Effort and value of all the apps in our sample (circles) compared to optimized portfolios.

Table 4. Suggested value-effort optimized portfolios.

SERVPORT	Effort	Value	#of M	#of O	#of A
{S1,S2,S3,S4,S5,S6,S7, S10,S11,S13,S14,S15, S19,S20,S21,S22,S23,S24,S25,S26, S28,S30,S31,S32, S35, S36}	626.5	8111	5	11	3
{S1,S2,S3,S4,S5,S6,S7, S10,S11, S14,S15, S16, S19,S21, S22,S23,S24,S25,S26, S28,S31,S32, S35, S36}	525.6	7432	4	10	3
{S1,S2,S3,S4,S5,S6,S7, S9, S10,S11,S13,S14,S15, S16,S19,S21, S22,S23,S24,S25,S26, S28,S32, S36}		7164	4	10	3
{S1,S2,S3,S4,S5,S6,S7, S10,S11, S14,S15, S16, S19,S20,S21, S22,S23,S24,S25,S26, S28,S30,S32, S35}		6907	5	10	3
{S1,S2,S3,S4,S5,S6,S7, S9, S10,S11,S13,S14,S15, S16,S19,S21, S22,S23,S25,S26, S28,S30,S32, S36}		6851	4	11	3
{S1,S2,S3,S4,S5,S6,S7, S9, S10,S11,S13,S14,S15, S16,S18,S19,S21, S22,S23,S25,S26, S28,S32, S35}		6818	4	10	3
{S1,S2,S3,S4,S5,S6,S7, S10,S14,S15,S19,S21, S22,S23,S25,S26, S28,S32, S35, S36}	452	6372	3	9	3
{S1,S2,S3,S4,S5,S6,S7, S9, S10, S11, S14,S15,S19,S21, S22,S23,S25,S26, S28,S32, S35}		6344	5	8	3
{S1,S2,S3,S4,S5,S6,S7, S9, S10, S14,S15,S16,S19,S21, S22,S23,S25,S26, S28,S32, S36}		6232	3	10	3
{S1,S2,S3,S4,S5,S6,S7,S9,S11,S16,S18,S22,S23,S24,S25,S26,S28,S31, S32,S36}	415.8	5733	2	9	3
{S1,S2,S3,S4,S5,S7,S9,S11,S16,S17,S18,S22,S23,S24,S25,S26,S28,S31, S32,S36}		5649	2	10	3
{S1, S2, S3, S4, S5, S6, S7, S9, S11, S16, S17, S18, S21, S22, S23, S24, S25, S32, S36}	341.5	5263	2	8	3
{S1, S2, S3, S4, S5, S6, S7, S9, S11, S16, S18, S22, S23, S24, S25, S28, S32, S36}		5096	2	9	3
{S1, S2, S3, S4, S5, S6, S7, S9, S11, S16, S18, S21, S22, S23, S24, S25, S28, S32, S36}		5055	2	9	3
{S1, S2, S3, S4, S5, S7, S9, S11, S16, S18, S21, S22, S23, S24, S25, S26, S32, S36}		5051	2	9	3
{S1, S2, S3, S4, S5, S7, S9, S11, S16, S17, S18, S22, S23, S24, S25, S28, S32, S36}		4780	2	9	3
{S1,S2,S3,S4,S5,S6, S7,S14, S19,S21, S22,S23,S25, S28,S32}	261	4506	1	6	3
{S1,S2,S3,S4,S5,S6, S7,S19,S21, S22,S23,S25, S28,S32}		4494	0	6	3
{S2,S3,S5,S7, S11,S19,S21, S22,S23, S25,S28,S32}	200	3836	1	6	2
{S1, S2,S3,S5,S7,S19,S21, S22,S23, S25,S28,S32}		3834	0	6	3
{S1, S2,S5,S7,S11, S19,S21, S22,S23, S25,S28,S32}		3804	1	6	2
{S1, S2,S3,S5,S7,S11, S19, S22,S23, S25,S28,S32}		3760	1	5	3
{S2,S3,S5,S7,S9, S11, S19, S21, S22,S23, S25,S28,S32}		3619	1	6	2
{S1,S3,S5,S11,S16,S21,S22,S23,S24,S25, S32,S36}	179.4	3461	2	6	2
{S1,S2,S3,S5, S11,S16,S18,S22,S23,S24,S25, S32}		3393	2	5	3
{S2,S3,S5,S7,S11,S16,S22,S23,S24,S25, S32}		3279	1	6	3
{S1,S2,S3,S5, S9,S11,S16,S22,S23,S24,S25, S32}		3184	1	5	2
{S1,S3,S5, S9,S11,S16,S22,S23,S24,S25, S32,S36}		3170	1	7	2
{S1,S3,S5,S9,S11,S16,S22,S23,S24,S25, S32}	150	2957	2	5	2
{S3,S5,S9,S11,S16,S21,S22,S23,S24,S25, S32}		2881	2	6	1
{S3,S5,S7, S21,S22, S23,S25, S32}	97.6	2602	0	4	1
{S1,S3,S5, S21,S22, S23,S25, S32}	86.6	2485	0	4	2
{S3,S4,S22,S23,S25, S32}	68.8	1935	0	3	1
{S3,S5,S11,S22,S25, S32}		1805	0	3	1
{S3,S5,S16,S22,S25, S32}		1726	1	2	1
{S3,S22,S24,S25, S32}		1570	0	3	1
{S16,S22,S23,S25, S32}		1542	0	4	0
{S5,S22,S25, S32}	38.5	1297	0	2	0
{S32}	8.5	353	0	0	1

5.6 Threats of validity

5.6.1 Service extraction

We relied on the app descriptions for service extraction even though it might not be the representation of all the included services. As the app description is considered as third degree data, we did not have any control over it [60]. Writing and describing apps including services is depending on the publishers' style. Some may completely describe services while the others point into the changes. This is also pointing to the problem of using third degree data because it was not originally developed with the intention to provide the services for our case study research [60] consequently, we do not claim that we cover all the services among all the apps but the ones that were stated by the publishers. Also we did not test the services to make sure that it was truly stated by the publishers [60]. To sum up, this paper only considered the claimed services of app providers which might differ from available services or may not cover all the actual services within apps.

Also having different languages and writing styles, publishers may call the same service functionality with different names. Using conceptual analysis, the tool extracted name concepts and composite concepts later we manually scanned the concepts and merge the ones which pointed into same service (i.e., VOD and Video on demand merged)

5.6.2 Crowdsourcing

Crowdsourcing provides access to complementary data, even though the micro task market which we utilized is restricting the access to personal information of users such as geographical data. Although the validation process is flexible, there are some concerns on the accuracy of responses. We design the task in a way to grant bonus to the high quality responses. For achieving more accurate results, we ran the same task, within the same time interval twice, one with bonus for high quality responses and 0.2$ incentive and the other, with 0.2$ incentive only. Our observations, shows more interactions toward bonus based task. We found the responses in the second sample more precise plus we receive emails including feedbacks on them. The comparison of results between the two approaches is shown in Figure 9. As discussed in Section 5.2.2 we hired 2 validation processes but still there is a possibility to have imprecise data from crowdsourcing which we were not detected and excluded from the data.

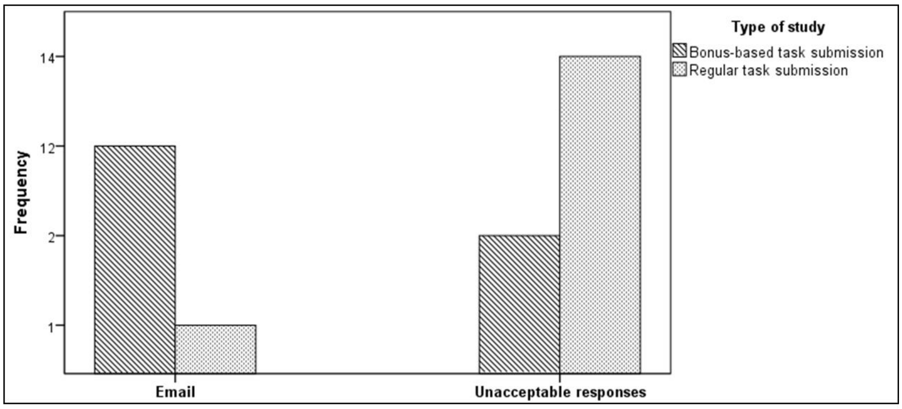

Figure 9. Comparison of responses from crowdsourcing without bonus (Regular) and bonus-based.

The other problem related to crowdsourcing is the question of how representative the crowd is. As the TV services are publicly available, the individuals responding in the crowdsourcing could be considered as the potential customers. The expertise of respondent involved in the crowdsourcing is of concern as well. While this is a general concern on crowdsourcing [62], in the context of our case study, we can highly expect that people has enough familiarity with TV services as these service is highly available publically.

5.6.3 Representativeness of value function

To confirm construct validity, we studied the representativeness of the customer satisfaction value function (10) used for our service portfolio optimization. For that, we performed two types of correlation analysis between (i) stated value and Android app market existing rate and (ii) stated value and number of services in must-be (M) ,one-dimensional (O) and attractive (A) Kano categories.

In order to evaluate the representativeness of calculated value based on Kano model, we analyze the correlation between calculated portfolio values for all the available apps in the market and the rate of them. For more precise analysis, we did this calculation (i) for all the apps in the market and (ii) for the market available Pareto services. In all these cases, we examined the following null hypothesis:

H_0: There is no correlation between the app rate and portfolio value

Comparison of all the apps: Calculating the Pearson's correlation resulted in a 0.285 correlation between rate and app value which is significant (sig = .000).

Comparison of correlation between market Pareto solutions: The calculation showed a 0.734 correlation between rate and app value with significance of 0.003 for the 14 outperformed apps in the market.
In the both investigations, the null hypothesis was rejected. The strength of correlation considering the optimized Pareto service portfolios is taken as an indication of the construct validity. The correlation among all 241 apps is positive and significant, but this is not very high. Dynamic and multi variable app market is affected by several known and unknown factors. Here, we only consider customer satisfaction as the value function. By this we showed that there is a significant relation between portfolio value and customer satisfaction indicated by rate although further studies are needed to define other influence factors as well.

In addition to the above, we also analyzed the Pearson correlation between number of one-dimensional, must-be and attractive services within each app and the app's market rate. This was done to observe the relation between number of service provided in each category and the rate of plans in the market. As presented in Table 5, the results do not show significant correlation between named Kano types and android market rate.

Table 5. Pearson's correlation between rate and number of M/O/A services within an app.

		rate	**Must-be services**	**One-dimensional service**
rate	Pearson Correlation	1	.053	.085
	Sig. (2-tailed)		.434	.184

Finally, we categorized the apps by the number of must-be (M) services and performed a One way ANOVA for the mean of rates. We did the same for the one-dimensional services as well. The ANOVA results are presented in Table 6 for must-be and one-dimensional services.

Table 6. One-way ANOVA test results of number of must-be and services and rate.

	Sum of Squares	**df**	**Mean Square**	**F**	**Sig.**
Must-be services	7.769	6	1.295	3.142	.006
One-dimensional services	3.241	11	.295	.669	.767

The correlation analysis in conjunction with ANOVA test shows no specific results to support the relation between the numbers of must-be and one-dimensional services and the rate of the apps. This could be observed in the mean plot of different number of must-be and one-dimensional services in Figure 10.

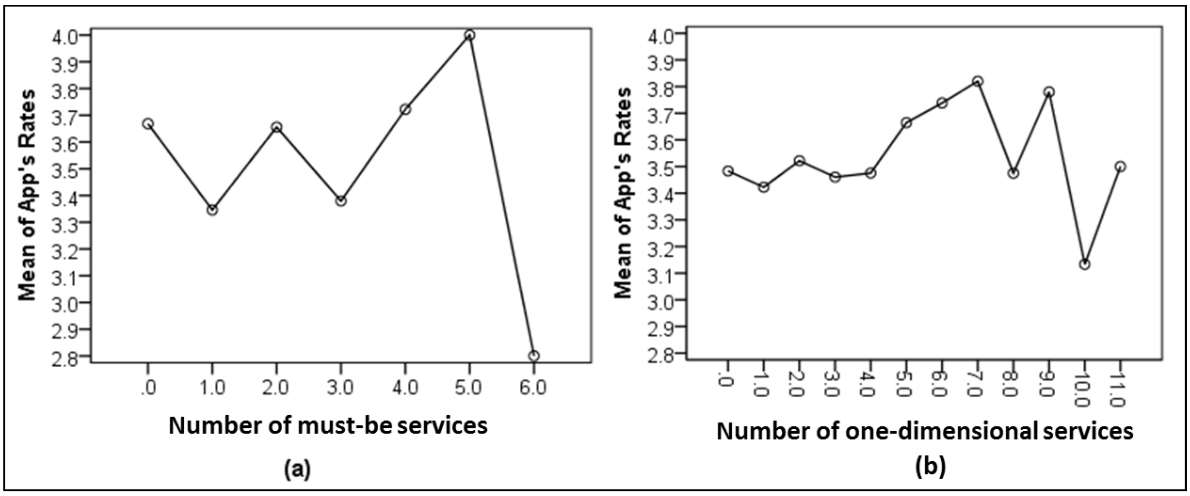

Figure 10. Mean plot of rate in dependence of the number of (a) *must-be* and (b) *one-dimensional* services.

6 Conclusion and Future Research

Advanced service portfolio planning is a cognitively and computationally complex problem. We have proposed a new approach for better understanding, formulating and algorithmically solving the problem of composing atomic services into a more valuable portfolio., with the increasing demand for continuous value delivery, a more systematic method than ad hoc is needed, one with the capability to offer optimized service portfolios based on most up-to-date information.

The Analytical Open Innovation (AOI) approach described in this paper is designed to address this need. AOI combines existing methods and techniques such as crowdsourcing, data mining and optimization in a new way to bring a high customer satisfaction value to the final product. Enlarging and improving the information used in planning, we have a better chance to address the right problem. We used the Kano questionnaire in conjunction with crowdsourcing for extracting customer value for each service. Service value was then defined based on the preference taken between the different Kano categories. As a result of the process, a set of optimized portfolio alternatives offered. These portfolios represent trade-off solutions balancing effort and customer value. The applicability of the approach was demonstrated by a case study from the app store market. For this application, the results represent optimized "super" apps which are based on extracting all types of atomic services from the existing (241) app descriptions, performing their crowdsourced Kano evaluation, doing expert-based effort estimation of atomic services, and finally running series of optimizations to determine the Pareto portfolio alternatives.

We consider the proposed research as being part of a more comprehensive research agenda. We foresee a wide range of future research challenges that needs to be addressed. One of them is to look more closely into service dependencies, in particular those ones corresponding to value synergies. Elicitation of these synergies and their reflection in the portfolio optimization process are two related topics in this context. Another one, is the real-world evaluation of the proposed service portfolios in the app store market. While there is evidence based on the Kano factors, we need to prove that providers and actual users are confirming this added value as well. This process should include the investigation of the impact of varying the weights describing the relative importance of the Kano categories. At the final stage of the whole planning and re-planning process, the AOI approach should support scenario-playing and other forms of varying other project parameters. Finally, the scalability of the entire approach needs to be investigated in order to assess its effectiveness and efficiency.

Acknowledgement

This research was partially supported by the Natural Sciences and Engineering Research Council of Canada, NSERC Discovery Grant 250343-12. Also we would like to thank Michael Felderer for discussions and all the valuable comments made. Also we would like to thank Navid Pourmomtaz for his comments on this paper. Last but not least, we acknowledge the reviewers comments and suggestions made related to the predecessor ICSOB 2014 paper.

References

1. Lovelock C, Wirtz J (2007) Services Marketing: People, Technology, Strategy. Upper Saddle River NJ: Prentice Hall

2. Strategic Research Agenda Need for Speed. (2013). http://www.digile.fi/Services/researchprograms/futureprograms.

3. Rometty VM (2013) A Conversation with Ginni Rometty. http://www.cfr.org/technology-and-science/conversation-ginni-rometty/p30181.

4. Enkel E, Gassmann O, Chesbrough H (2009) Open R&D and open innovation: exploring the phenomenon. R and D Management 39 (4):311-316

5. Chesbrough H (2003) Open Innovation: The New Imperative for Creating and Profiting from Technology. Harvard Business Press, Boston, Massachusetts

6. Marjanovic S FC, Joanna C (2012) Crowdsourcing based business models: In search of evidence for innovation 2.0. Science and Public Policy 39 (3):318-332

7. Cleland-Huang J, Horatiu Dumitru, Chuan Duan, and Carlos Castro-Herrera (2009) Automated Support for Managing Feature Requests in Open Forums. Communications of the ACM 52 (10):68-74

8. Rittel HW, Webber MM (1973) Planning Problems Are Wicked Problems. Developments in Design Methodology:155 - 169

9. Ritchey T (2011) Wicked Problems--Social Messes: Decision Support Modelling with Morphological Analysis. (Risk,Governance and society), Springer

10. Ngo-The A, Ruhe G (2008) A systematic approach for solving the wicked problem of software release planning. Soft Computing 12 (1):95-108

11. Du G, Richter MM, Ruhe G (2005) An explanation oriented dialogue approach for solving wicked planning problems. In: AAAI Fall Symposium pp 62-75

12. Heikkilae V, Jadallah A, Rautiainen K, Ruhe G (2010) Rigorous support for flexible planning of product releases - A stakeholder-centric approach and its initial evaluation. Proceedings 43rd Hawaii International Conference on System Sciences, pp.1-10

13. Wiegers K (2003) Software Requirements. Microsoft Press

14. Carlshamre P (2001) An Industrial Survey of Requirements Interdependencies in Software Product Release Planning. Fifth International Symposium on Requirements Engineering (RE'01):84-91

15. Nayebi M, Ruhe G (2014) An open innovation approach in support of product release decisions. In: Proceedings of the 7th ACM International Workshop on Cooperative and Human Aspects of Software Engineering, pp 64-71

16. Nayebi M, Ruhe G (2014) Analytical Open Innovation for Value-Optimized Service Portfolio Planning. Proceedings ICSOB 2014, pp 273-288

17. Agarwal N, Karimpour R, Ruhe G (2014) Theme-based Product Release Planning: An Analytical Approach.

18. Khurum M, Gorschek T, Wilson M (2013) The software value map—an exhaustive collection of value aspects for the development of software intensive products. Journal of Software: Evolution and Process 25 (7):711-741

19. Sharp H, Finkelstein A, Galal G Stakeholder (1999) Identification in the Requirements Engineering Process. In: 10th International Workshop on Database and Expert Systems Applications, pp 387-391

20. Gorschek T, Fricker S, Palm K, Kunsman S (2010) A lightweight innovation process for software-intensive product development. IEEE Software 27 (1):37-45

21. Fitsilis P, Gerogiannis V, Anthopoulos L, Savvas IK (2010) Supporting the requirements prioritization process using social network analysis techniques. In: Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE), 19th IEEE International Workshop on, 2010. pp 110-115

22. Shepperd M (2007) Software project economics: a roadmap. In: Future of Software Engineering, 2007. FOSE'07, 2007, pp 304-315

23. Boehm B, Valerdi R (2011) Impact of software resource estimation research on practice: a preliminary report on achievements, synergies, and challenges. In: Proceedings 33rd ICSE , pp 1057-1065

24. Carman JM, Langeard E (1980) Growth Strategies for Service Firms. Strategic Management Journal 1 (1):7-22

25. Chanal VC, Marie-Laurence (2008) How to invent a new business model based on crowdsourcing the crowdspirit. Paper presented at the Conférence de l'Association Internationale de Management Stratégique

26. Rohrbeck R, Hölzle K, Gemünden HG (2009) Opening up for competitive advantage–How Deutsche Telekom creates an open innovation ecosystem. R&d Management 39 (4):420-430

27. Jansen S, Brinkkemper S, Souer J, Luinenburg L (2012) Shades of gray: Opening up a software producing organization with the open software enterprise model. Journal of Systems and Software 85 (7):1495-1510

28. Conboy K, Morgan L (2011) Beyond the customer: Opening the agile systems development process. Information and Software Technology 53 (5):535-542

29. A. Majchrzak, Malhotra A (2013) Towards an information systems perspective and research agenda on crowdsourcing for innovation. Journal of Strategic Information Systems 22.4: 257-268.

30. Brabham DC (2008) Crowdsourcing as a model for problem solving: An introduction and cases. Convergence 14 (1):75-90

31. Cleland-Huang J, Dumitru H, Duan C, Castro-Herrera C (2009) Automated support for managing feature requests in open forums. Communications of the ACM 52 (10):68-74

32. Estelle´s Arolas E (2012) Towards an integrated crowdsourcing definition. Journal of Information Science 38 (2):189–200

33. Doan AR, Raghu. Halevy, Alon Y (2011) Crowdsourcing systems on the World-Wide Web. Communications of the ACM 54 (4):86-96

34. Estellés-Arolas E, González-Ladrón-de-Guevara F (2012) Towards an integrated crowdsourcing definition. Journal of Information science 38 (2):189-200

35. Gregory D. S, Onook O, Rajiv K (2013) Rules of Crowdsourcing: Models, Issues, and Systems of Control. Information Systems Management 30 (1):2-20

36. Ridder PD (2013) Open Innovation & Crowdsourcing Examples. http://www.boardofinnovation.com/list-open-innovation-crowdsourcing-examples/.

37. Luor T, Lu H-P, Chien K-M, Wu T-C (2012) Contribution to quality research: A literature review of Kano's model from 1998 to 2012. Total Quality Management & Business Excellence (ahead-of-print):1-14

38. R.P.L. Buse, T Zimmerman (2012) Information Needs for Software Development Analytics. Paper presented at the Software Engineering (ICSE), 2012 34th International Conference on, June 2-9

39. Wnuk K, Runeson P (2013) Engineering Open Innovation–Towards a Framework for Fostering Open Innovation. In: The Journal of Strategic Information Systems. From Physical Products to Software Services and Solutions. Springer, pp 48-59

40. West J, Bogers M (2013) Leveraging external sources of innovation: a review of research on open innovation. Journal of Product Innovation Management, Vol. 31 No. 4, pp. 814-31

41. Berntsson Svensson R, Lindberg Parker P, Regnell B (2011) A prototype tool for QUPER to support release planning of quality requirements. In: Fifth International Workshop on Software Product Management, pp 57-66

42. Regnell B, Svensson RB, Olsson T (2008) Supporting roadmapping of quality requirements. IEEE Software 25 (2):42-47

43. Sauerwein E, Bailom F, Matzler K, Hinterhuber HH (1996) The Kano model: How to delight your customers. In: International Working Seminar on Production Economics, pp 313-327

44. Kano N, Seraku N, Takahashi F, Tsuji S (1984) Attractive quality and must-be quality. Journal of the Japanese Society for Quality Control 14 (2):147-156

45. Akao Y (2004) Quality function deployment, Cambridge, MA: Productivity Press.

46. Mikulic J, Prebežac D (2011) A critical review of techniques for classifying quality attributes in the Kano model. Managing Service Quality 21 (1):46-66

47. Xu Q, Jiao RJ, Yang X, Helander M, Khalid HM, Opperud A (2009) An analytical Kano model for customer need analysis. Design Studies 30 (1):87-110

48. Chen C-C, Chuang M-C (2008) Integrating the Kano model into a robust design approach to enhance customer satisfaction with product design. International Journal of Production Economics 114 (2):667-681

49. Lee Y-C, Huang S-Y (2009) A new fuzzy concept approach for Kano's model. Expert Systems with Applications 36 (3):4479-4484

50. Kano N, Seraku N, Takahashi F, Tsuji S (1996) Attractive quality and must-be quality. The Best on Quality 7:105

51. MTurk (2013) MTurk. https://www.mturk.com/mturk/.

52. Very Best Choice light, Expert Decisions Inc. http://edi.lite.verybestchoice.com:3000/

53. Berelson B (1952) Content analysis in communication research, The Free Press, Glencoe, Illinois.

54. Zhang D, Han S, Dang Y, Lou J, Zhang H, Xie T (2012) Software Analytics in Practice. IEEE Software 30 (5)

55. Ruhe G (2010) Product Release Planning: Methods, Tools and Applications. CRC Press,

56. www.releaseplanner.com ReleasePlanner®, www.releaseplanner.com. Expert Decisions Inc. Accessed March 2008

57. De Boever J, De Grooff D (2010) Peer-to-Peer Content Distribution and Over-The-Top TV: An Analysis of Value Networks. In: Handbook of Peer-to-Peer Networking. Springer, USA, pp 961 - 983

58. Montpetit M-J, Klym N, Mirlacher T (2011) The future of IPTV (connected, mobile, personal and social). Multimedia Tools and Applications 53 (3):519-532

59. Harman M, Jia Y, Zhang Y (2012) App store mining and analysis: MSR for app stores. In: Mining Software Repositories (MSR), 2012 9th IEEE Working Conference on, pp 108-111

60. Runeson P, Höst M (2009) Guidelines for conducting and reporting case study research in software engineering. Empirical software engineering 14 (2):131-164

61. Martello S, Torth P (1990) Knapsack Problems, Algorithms and Computer Implementation. NewYork: Wiley,

62. Lee K (2013) Combating threats to the quality of information in social systems. Texas A & M University,