Research Center
Finance & Information Management

Fraunhofer FIT

Project Group
Business & Information
Systems Engineering

Assessing the Criticality of IT Projects in a Portfolio Context using Centrality Measures

by

Thomas Wolf

presented at: 12th International Conference on Wirtschaftsinformatik (WI), Osnabrück, Germany, 2015

University of Augsburg, D-86135 Augsburg
Visitors: Universitätsstr. 12, 86159 Augsburg
Phone: +49 821 598-4801 (Fax: -4899)

University of Bayreuth, D-95440 Bayreuth
Visitors: Wittelsbacherring 10, 95444 Bayreuth
Phone: +49 921 55-4710 (Fax: -844710)

www.fim-rc.de

WI-481

Assessing the Criticality of IT Projects in a Portfolio Context using Centrality Measures

Thomas Wolf[1]

[1] University of Augsburg, FIM Research Center, Germany

thomas.wolf@fim-rc.de

Abstract. Recent technological developments associated with changes in customer expectations have required continuous innovation from companies all over the world, thereby driving these companies' IT portfolios towards increasing complexity and interdependency. Simultaneously, existing methods of IT portfolio management are not able to cope with this interconnectedness of IT projects, and too little research has been performed on appropriate risk assessments of dependency structures. By considering such dependency structures as IT project networks, we draw on centrality measures to assess the risk associated with inherent project dependencies. We examine different kinds of centrality measures, whether and to what extent they are able to depict characteristics specific to IT project networks. Based on the most appropriate measure, we derive criticality values indicating projects crucial to the IT portfolio's success. These criticality values should empower companies to successfully manage their IT portfolio.

Keywords: IT Projects, IT Portfolio, Dependencies, Centrality, Criticality

1 Introduction

Information technology (IT) has become a critical success factor in many industries. However, despite various planning techniques, there is still a huge number of failed IT projects. In this context, the "chaos report" is often quoted, which states that 80% of all IT projects are only partly implemented or even fail completely [1]. Moreover, FLYVBERG and BUDZIER contend that around 16% of IT projects cause on average budget deficits of about 200% [2]. A questionnaire by the RADAR GROUP, surveying 560 IT decision makers in Scandinavia, concludes that one reason for IT project failure is a lack of transparency regarding dependencies [3]. Since IT projects usually are not accomplished in isolation or pairwise but rather within an aggregated portfolio of several IT projects, they incorporate higher-order dependencies [4]. This becomes even more relevant, as recent technological developments and associated changes in customer expectations force companies to continuously come up with innovations [5]. Consequently, IT projects which previously would have been developed as one coherent solution, are now split into several standalone but interrelated IT services with customer impact, to satisfy the continuous demand for innovation. Therefore, IT project portfolios, henceforth simply referred to as "IT portfolios", tend to comprise many

12th International Conference on Wirtschaftsinformatik,
March 4-6 2015, Osnabrück, Germany

small projects rather than a few big ones. This further heightens the need from praxis for a more detailed assessment of risk due to related dependencies.

In addition, literature considers the appropriate assessment of dependencies as a crucial risk during the project-planning phase [6]. Although KUNDISCH and MEIER assert that compared to the claimed importance of this topic, relatively little research can be found [7], there are at least some approaches of IT project and portfolio management that tried to incorporate dependencies to some extent (e.g. [8-11]). However, existing methods based on classical portfolio theory are not sufficient to cope with characteristics specific to IT portfolios [12]. Since the structure of dependencies between projects in an IT portfolio is important for the success of each single IT project [13], each single project can also be crucial to the overall success of the portfolio. This is known as systemic risk and is characteristically based on direct and indirect dependencies within network structures. Therefore, we consider IT portfolios as IT project networks, and present a novel approach drawing on concepts from sociological research instead of classical portfolio theory. By considering projects of an IT portfolio as nodes and dependencies amongst them as arcs, we can analyze the corresponding network based on centrality measures, derived from the mathematical field of graph theory, and strive to identify the most important node of the network [14]. Projecting this onto IT portfolios we consequently aim to identify the most critical project of the IT project network. Therefore, we set forth the following research question: *"Can centrality measures be used to assess the criticality of a project to its corresponding IT portfolio, based on inherent project dependencies?"*

To answer this question, we assess different kinds of common centrality measures and outline whether and to what extent they can depict characteristics specific to IT portfolios, in order to consider them appropriate. By determining which projects are crucial to the success of the overall IT portfolio, the results should empower companies to take appropriate actions (e.g. reallocation of dedicated resources) in order to successfully manage their IT portfolio. MEREDITH ET AL. [15] proposed a three-stage research cycle for activities in the field of operations research. They cluster research into description, explanation and testing phase. Our research is located in the explanation stage of this cycle, which is supposed to yield first concepts and models from which causal relationships and testable hypotheses can be derived.

The remainder of the paper is organized as follows: Section 2 presents a literature review of different kinds of dependencies in IT portfolios and their current assessment. Section 3 outlines the basic principles of the approach, including preliminary considerations and an application example to facilitate comprehensibility and verify applicability. Finally, Section 4 summarizes, concludes, and depicts the limitations of the approach.

2 Literature Review

To develop a novel method that properly assesses dependencies and contributes to existing literature, it is necessary to know which kinds of dependencies exist in IT portfolios and how they are currently appraised. Therefore, a keyword (*dependency*, *interdependency*, *interaction*, *project*, *portfolio*, *information technology*, *information systems*, *model*, *approach*, *quantification*, *assessment*) based search of different data bases (*AIS Electronic Library*, *EBSCOhost*, *EmeraldInsight*, *ProQuest*, *ScienceDirect*, *Wiley*) was conducted. Since not each database supports the same AND/OR conjunction of search terms, in some cases the search term has been adapted. To account for different methods and approaches assessing dependencies in varying disciplines, the search term has to be kept at a generic level. Consequently, the resulting set of articles is too large to directly process it. To condense the number of articles we stick to the approach of KUNDISCH and MEIER [7], including only articles being published in the top journals of the *Information Systems, Production and Operations Management*, and *Project Management* disciplines. Subsequently, the articles' titles were conducted to decide whether an article contributes to the research objective or not. If the title did not suffice to decide whether the article properly contributes to the topic, the abstract was examined. By analyzing the articles it became apparent, that some of them, despite the initial impression, did not properly contribute to the research objective and hence had to be excluded afterwards. To complete the search procedure, we did a forward and backward search of citations in the set of relevant articles, like recommended by WEBSTER and WATSON [16].

Based on this investigation we can constitute that in existing IT portfolio literature, there are different kinds of dependencies connecting two or more projects. While some articles just mention certain types of dependencies, others try to introduce whole frameworks, structuring different categories of dependencies based on specific characteristics. Like SANTHANAM and KYPARISIS, LEE and KIM, TILLQUIST ET AL. or ZULUAGA ET AL. most articles in literature present either some or all of the following dependencies: resource, technical, and benefit dependencies [8, 17-19]. Generally, resource dependencies refer to projects competing for any kind of resources. Technical dependencies most commonly refer to projects competing for technical systems or applications [17]. However, technical systems and applications can also be considered as input resources of a project. Therefore, WEHRMANN subdivides resource dependencies into personnel and technical dependencies [9]. In contrast, KUNDISCH and MEIER introduce a framework subdividing resource dependencies into allocation, performance, and sourcing interactions [10]. Benefit dependencies are also considered as synergies, and can be realized if the benefit of one or more projects increases while being simultaneously implemented with another project. One example could be the reuse of code fragments for two similar software development projects. For further explanations and differentiations of synergies, refer to [12].

Structuring dependencies by characteristics, WEHRMANN ET AL. and ZIMMERMANN differentiate between inter-temporal and intra-temporal dependencies [9, 13]. Inter-temporal dependencies refer to projects taking place at different points in time; for example, if a project is based on a preceding one. Intra-temporal dependencies refer to different projects taking place at the same point in time; according to WEHRMANN ET AL., they involve structural dependencies, which refer to projects that are based on the same processes, IT functionalities or data, and resource dependencies [9].

Determining how and to what extent dependencies between different projects exist is a topic most commonly left to expert judgment. For such evaluations, scoring systems are often the method of choice (cf. [20-22]). However, the processing of resulting values, henceforth considered as dependencies, is handled differently. While most models in the context of IT portfolio management incorporate dependencies within the risk assessment, there are also some different approaches. Based on the differentiation between intra- and inter-temporal dependencies [9], we therefore subsequently briefly depict how current methods of IT portfolio management consider dependencies.

To account for intra-temporal dependencies, SANTHANAM and KYPARISIS propose a non-linear optimization model, considering resource and technical dependencies as auxiliary conditions to their objective function of selecting an optimized project portfolio based on fixed budgets [17]. Further approaches considering dependencies as auxiliary conditions in an optimization model can be found in works by Lee and Kim and KUNDISCH and MEIER [8, 10]. Considering dependencies in terms of risk, e.g. BUTLER ET AL., WEHRMANN ET AL. and BEER ET AL. refer to portfolio theory [23] to determine a risk and return optimized IT portfolio [9, 11, 24]. They consider dependencies by correlation coefficients based on covariances of the corresponding IT projects. VERHOEF introduced a modified discounted cash flow method, which evaluates dependencies implicitly while focusing on cost and time risks within the interest rate [25]. Since many of the existing approaches incorporating intra-temporal dependencies consider only dependencies between two different projects or depict them predominantly by financial restrictions, they partially fall short [26]. Furthermore, some approaches are adopted from financial methods. Therefore, they would have to fulfill specific premises (e.g. portfolio theory), which are however not at all or only partially applicable in the context of IT portfolios. Other methods again feature a very high level of subjectivity (e.g. scoring methods) since they are almost purely based on expert estimations.

Inter-temporal dependencies are most commonly considered based on real option models. In this context, many approaches have been proposed (cf. [27-30]) using either the Black-Scholes model or binomial trees. Since these methods are derived from financial option methods and have been adapted to real options, they are considered somewhat inappropriate for evaluation of inter-temporal dependencies in an IT project portfolio context, due to their underlying premises [31, 32]. For a more detailed discussion on whether restrictive premises of financial option methods can be adapted to real options and whether the models can be appropriately used in this context, please refer to [33, 34].

Based on the previous examination of current methods for IT portfolio evaluation, we can conclude that existing approaches cannot be considered completely appropriate regarding incorporation of dependencies prevailing in IT project networks. Besides, the most important drawback is, to the best of our knowledge, none of the existent IT portfolio management techniques explicitly accounts for transitive dependencies between IT projects. However, t an assessment of these transitive dependencies is crucial to an appropriate risk assessment in network-like structures.

3 Model

Concepts from the sociological research field of social network analysis have recently been applied to several other research areas, such as supply chain management, logistics, and IT landscape management, in order to assess risk originating from dependencies within these network structures (cf. [35-37]). We interpret IT portfolios as IT project networks, by considering projects as nodes and dependencies amongst them as arcs. Consequently, we can evaluate the adaption of social network measures to the research area of IT portfolios by analyzing the appropriateness of different centrality measures, in order to assess the risk of the portfolio's corresponding IT project network. Centrality measures strive to identify the most important node of a network [14]. For IT project networks, we henceforth assume that the centrality values of nodes represent criticality values of projects, indicating their importance to the success of the corresponding IT portfolio, based on the projects' dependencies. Furthermore, we define the success of an IT portfolio as its accomplishments, time- and budget-wise.

3.1 Modeling IT Portfolios as Networks

The mere assertion that two projects of an IT portfolio are somehow dependent is not sufficient for a company to allocate resources adequately. To do so, the company needs information on the direction of this dependency. Therefore, the IT project network of a corresponding IT portfolio can be visualized as based on directed arcs. An arc pointing from one project to another indicates a dependency of the initiating project (where the arc originates) on the project where the arc ends. To assess the criticality of a project, we consequently focus on incoming instead of outgoing arcs. Furthermore, dependencies between different projects are rarely equally weighted in reality. Ergo, to account for different strengths of dependencies, we presuppose the arcs of an IT project network to be weighted. However, the calculation of such bilateral dependency weightings does not fall under the scope of this paper, as we rather focus on how to assess the coherence of these identified dependencies within a network environment. We therefore assume that it is possible to quantify any kind of dependencies for pairwise combinations of IT projects. The validity of this assumption is borne out in theory, as corresponding quantification techniques based on expert judgments and scoring models are already used in the field of IT portfolios (cf. [20-22]).

Before we are able to identify crucial projects by deriving criticality values based on the projects' dependencies, we first need to examine whether and which centrality measures are appropriate to account for the specific characteristics of IT project networks.

3.2 Requirements to Centrality Measures in IT Project Networks

Since the only prerequisite for the application of centrality measures is the existence of a network composed of nodes and arcs, such measures nowadays are widely applied, although most of them were originally introduced in the social network context [38]. However, like social (cf. [39]) or supply networks (cf. [36]), IT project networks feature specific characteristics that must be considered in order to properly assess the projects' criticality. These characteristics are based on a common understanding of dependencies in IT portfolios. We subsequently outline the underlying logical consideration and derive some simple and generic requirements which a centrality measure must take into account in order to be considered reasonably applicable in the IT portfolio context and in the context of this paper. However, the derived requirements can rather be considered as minimum requirements than as a comprehensive list, and do not feature any kind of prioritization.

There are centrality measures that have been designed for either directed or undirected networks. However, with slight modifications, many of them can be applied to both directed and undirected networks. As explained above, we can visualize IT project networks as composed of directed arcs. Consequently, an appropriate centrality measure should account for directed relations as stated in the following requirement:

Requirement (Req.) 1: The measurement accounts for directed relations between projects.

Furthermore, we consider four influential factors in order to determine the importance of an IT project to its corresponding portfolio: The strength of the dependencies (a), the number of directly dependent projects (b), the number of indirectly dependent projects (c), and the inherent importance of directly and indirectly dependent projects (d).

Regarding (a), we assume an IT project to be more important if it has strong dependencies to other projects, as opposed to the case where these dependencies are weak. By considering the arcs of an IT portfolio to represent corresponding dependencies, the strength of dependencies can be depicted by weighted arcs. Accordingly, the criticality value should increase with the weighting of arcs or rather the strength of dependencies, as stated in the following requirement:

Requirement (Req.) 2: The result of the measurement for a specific project increases with the strength of relations to dependent projects.

Regarding (b), we expect a project to be more important to its corresponding network if there are many other projects in the network that directly depend upon it. For example, a single project is more important to its corresponding portfolio if it has five other projects which directly depend on it, in contrast with the case where it has just three others directly dependent on it. Assuming arcs represent dependency relations, an appropriate measure should hence consider that the criticality value of a single project increases with the number of relations pointing directly from other projects of the network towards it. This is stated in the following requirement:

Requirement (Req.) 3: The result of the measurement for a specific project increases with the number of directly dependent projects.

Regarding (c), we expect a project to influence the criticality of another project, even though it does not directly but rather indirectly depend upon the other project. Extending the example from above, a single project that has only three directly dependent projects is not necessarily less important than the project which has five directly dependent projects. The importance does not solely depend on the number of directly dependent projects, but also on the number of indirectly dependent projects. Consequently, an appropriate measure should also consider that the criticality of a project increases with an increasing number of indirectly or transitive dependent projects, as stated in the following requirement:

Requirement (Req.) 4: The measurement accounts for transitive dependencies, as the result increases with the number of indirectly dependent projects.

Regarding (d), we additionally expect a project to further influence the importance of another project it depends on, if it has a high importance itself. This means that a project with a dependent project ranked as important has a higher importance to the network itself, as opposed to a project having a relatively unimportant dependent project. Consequently, an appropriate measure should also consider that the criticality of a node with a higher criticality on its own contributes more to the criticality of another node it is dependent on, rather than a node with a lower criticality. This is stated in the following requirement:

Requirement (Req.) 5: The result of the measurement of a specific project increases with the importance of directly and indirectly dependent projects.

Although there are many different centrality measures, we in a first step will only introduce some of the most common ones in the following. In particular we will examine, how and to what extent they account for Req. 1-5, and if they can reasonably be applied in the IT project network context.

3.3 Examination of Different Centrality Measures

Closeness centrality is a centrality measurement that determines the importance or status of a node in a network based on how close a node is to the others in a network [14]. The calculation therefore is based on the summed distances of one node e from all other $n-1$ nodes of the network. Considering $d(e,i)$ to represent the shortest path from node e to any other node i, the closeness centrality $C_C(e)$ can be calculated [40]. However, since $C_C(e)$ is dependent on the overall number n of nodes in the network, we can derive a corresponding standardized $\overline{C_C}(e)$ [14]. Both $C_C(e)$ and $\overline{C_C}(e)$ are depicted in the following equation:

$$C_C(e) = \frac{1}{\sum_{i=1}^{n} d(e,i)}, \qquad \overline{C_C}(e) = \frac{n-1}{(\sum_{i=1}^{n} d(e,i))} \quad with\ i \neq e \tag{1}$$

This measure is applicable to directed and undirected networks and thus fulfills Req. 1. It also accounts for weighted arcs, which in this case represent distances between adjacent nodes. Since short distances are advantageous for the purpose of closeness centrality, the measurement increases for declining strength of weights and therefore does not fulfill Req. 2. Moreover, it falls short on Req. 3-5, since it neither increases with the number nor the criticality of directly or indirectly dependent projects.

Another measure of centrality, determining the status of a node by how often it is located on the shortest path between all other pairwise combinations of nodes, is ***betweenness centrality***. Assuming p_{ij} to be the number of shortest paths connecting any node i and j, and $p_{ij}(e)$ the number of paths containing node e, the betweenness centrality $C_B(e)$ can be calculated [41]. Since also $C_B(e)$ is dependent on the overall number n of nodes in the network, we can derive a corresponding standardization $\overline{C_B}(e)$ as well [14]:

$$C_B(e) = \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{p_{ij}(e)}{p_{ij}}\ , \qquad \overline{C_B}(e) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{p_{ij}(e)}{p_{ij}}}{\frac{(n-1)\cdot(n-2)}{2}} \tag{2}$$

$$with \quad i \neq e, \qquad j \neq e, \qquad j < i$$

Although this measure has been developed specifically for undirected relations, GOULD has shown that it can also be used for directed relations based on geodesics between pairs of entities [42]. Therefore, it fulfills Req.1. However, this measure considers transitive dependencies and thus can be used to analyze which projects are connected over several stages of the IT project network, it does not fulfill Req. 3 and 4, as it does not increase with the number of transitive or directly dependent projects. It also falls short on Req. 2 and 5, since it does not increase for the strength of dependencies or the criticality of dependent projects.

Degree centrality can be calculated based on the number of arcs directly connecting one node of a network to the others. The existence of connections between nodes in the network is depicted in a so-called "adjacency matrix", which consequently represents the network structure. This adjacency matrix ***A*** in the simplest case contains binary elements a_{ij} with $a_{ij} = 1$ if there is a relation between node $i = 1 \ldots n$ and node $j = 1 \ldots n$ and $a_{ij} = 0$ if not. By considering the number of nodes a specific node e is linked to, the degree centrality $C_D(e)$ can be calculated. To enable comparability for different network sizes, a standardized measure $\overline{C_D}(e)$ has been proposed similar to closeness and betweenness centrality [14, 38]:

$$C_D(e) = \sum_{i=1}^{n} a_{ie}\ , \qquad \overline{C_D}(e) = \frac{\sum_{i=1}^{n} a_{ie}}{n-1} \tag{3}$$

By distinguishing between in- and out-degree centrality relating to incoming and outgoing arcs of a node, this measurement is applicable to directed networks and therefore fulfills for Req. 1. Since the measure also increases with the number of directly dependent projects and the strength of dependencies, it also fulfills Req. 2 and 3. However, degree centrality does not account for transitive dependencies and thus does not fulfill Req. 4. It also falls short on Req. 5 since it does not increase with the importance of dependent projects.

In order to account for the phenomenon that more interconnected nodes contribute more strongly to the status of nodes to which they are adjacent, other centrality measures such as the ***eigenvector centrality*** have been developed [43]. Assuming $\boldsymbol{v} = (v_1, \dots, v_n)^T$ to be an eigenvector for the maximum eigenvalue $\lambda_{max}(\boldsymbol{A})$ of the adjacency matrix $\boldsymbol{A}$, the eigenvector centrality $C_E(e)$ for a node e is defined as follows [43]:

$$C_E(e) = v_e = \frac{1}{\lambda_{max}(\boldsymbol{A})} \cdot \sum_{j=1}^{n} a_{je} \cdot v_j \tag{4}$$

With $\boldsymbol{A}^T$ being the transposed matrix of the adjacency matrix $\boldsymbol{A}$, the respective matrix representation of equation (4) can be derived:

$$\boldsymbol{A^T x} = \boldsymbol{x} \tag{5}$$

Eigenvector centrality quantifies to which extent nodes are related to others within the same network [43]. For each node that depends upon another, it weights the corresponding binary value in the adjacency matrix $\boldsymbol{A}$ by the eigenvector centrality of the dependent one. When this concept is applied to IT project networks, the binary value, indicating whether a project i is dependent on another project j, is weighted by the criticality value of project j. Since this measure has been developed for directed networks, it fulfills Req. 1. It also fulfills Req. 2, since it increases with the strength of dependencies. By calculating eigenvector centrality for a specific node, the value theoretically also increases with the number of directly and indirectly dependent nodes, as well as with their criticality. However, as the status of a node is solely influenced by its relations to other nodes, this method has a major drawback: If a node has no incoming relations from others, its status equals 0 and it therefore does not contribute to the importance of other nodes [43]. Therefore, this measure in fact fulfills Req. 5, but falls short on Req. 3-4.

To account for this drawback, eigenvector centrality has been further enhanced and some derivatives have evolved. One of these derivatives introduced by BONACICH and LLOYD, is ***alpha centrality*** [43]:

$$\boldsymbol{x} = (\boldsymbol{I} - \propto * \boldsymbol{A}^T)^{-1} * \boldsymbol{e} \tag{6}$$

This centrality measure overcomes the mentioned drawback of eigenvector centrality by assigning an exogenous status to each node of the network. In equation (6) this exogenous status is represented by the vector $\boldsymbol{e}$. This vector theoretically enables the ability to account for influences like project budged, which determine the exogenous status of different nodes to different extents. However, as the assessment of a project´s exogenous status is not in scope of this paper but rather a topic for further research, we stick to the work of BONACICH and LLOYD, who exemplarily considered $\boldsymbol{e}$ as a vector of ones [43]. Consequently, the initial (exogenous) status of each node of the network is set to 1, independent of its relations to other nodes. The remaining elements of the equation are the identity matrix $\boldsymbol{I}$, the transposed adjacency matrix $\boldsymbol{A}^T$ and the scalar $\propto > 0$, representing a ratio for the relative relations between the exogenous (assigned) and endogenous (inherent) status of the nodes. Consequently, if $\propto$ is close to its lower boundary

0, the corresponding centrality values are close to the exogenous status of the nodes. In contrast, if $\propto$ is close to its upper boundary $\frac{1}{\lambda_{max}(\boldsymbol{A})}$, where $\lambda_{max}(\boldsymbol{A})$ represents the maximum eigenvalue of $\boldsymbol{A}$, the corresponding centrality values are almost exclusively based on the endogenous status, or rather on the network or relation structure.

Since this measurement in contrast to eigenvector centrality indeed increases with the number of each directly and indirectly dependent node, it not only accounts for Req. 1, 2, and 5, but also for Req. 3 and 4. Moreover, it features the possibility of including exogenous influences like project size or volume. Therefore, we consider this measure as appropriate for the criticality assessment of projects in the sense of this paper. Table 1 summarized the results regarding the appropriateness of the five requirements for all examined centrality measures in this section.

Table 1. - Examination of centrality measures - summary

Centrality Measure	*Req. 1*	*Req. 2*	*Req. 3*	*Req. 4*	*Req. 5*
Closeness Centrality	✓	-	-	-	-
Betweenness Centrality	✓	-	-	-	-
Degree Centrality	✓	✓	✓	-	-
Eigenvector Centrality	✓	✓	-	-	✓
Alpha Centrality	✓	✓	✓	✓	✓

3.4 How to Assess Critical Projects based on Alpha Centrality

Presuming that dependencies between IT projects can be quantified, and considering these to equal network alike structures, alpha centrality allows the derivation of an interpretable criticality value indicating an individual project's importance to the overall success of the IT project network. In doing so, it not only accounts for direct dependencies, like the number of directly dependent projects, but also for indirect or transitive dependencies. To facilitate the comprehensibility and illustrate the suitability of alpha centrality in an IT portfolio context, this section briefly introduces the basic principles of the measurement using the three simple topology examples shown in Figure 1.

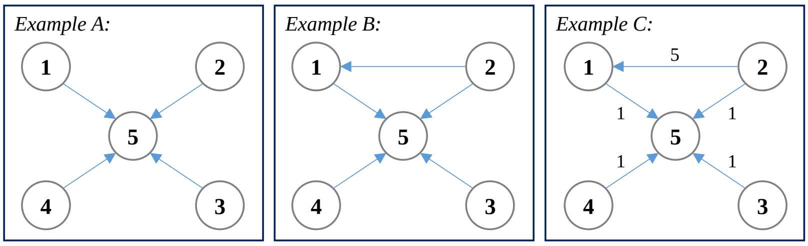

Fig. 1. – Examples of simple IT project network topologies

Alpha centrality uses an $n \times n$ adjacency matrix $\boldsymbol{A}$ whose elements a_{ij} with $i, j = 1 \dots n$ represent the connections of the network and consequently the projects' dependencies. Considering arcs as unweighted, each element a_{ij} represents a binary value indicating whether project i is directly dependent on project j or rather whether i contributes to the criticality of j. In the case of weighted arcs, each element a_{ij} represents the weight of the corresponding dependency relation between i and j.

In the following, the binary adjacency matrices of example A and B are depicted, as well as the weighted adjacency matrix of example C.

$$\boldsymbol{A}_A = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{A}_B = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{A}_C = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 5 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

In example A, project 1, 2, 3, and 4 depend on project 5. Consequently, one would assume the latter as most important, or rather most critical to the success of the overall IT portfolio and therefore has the highest alpha centrality value. Assuming $\boldsymbol{e} = [1, 1, 1, 1, 1]$ to be a vector of ones and based on the corresponding adjacency matrix $\boldsymbol{A}_A$, we can calculate the alpha centrality vector $\boldsymbol{x}^{\boldsymbol{T}} = [1, 1, 1, 1, (1 + 4 \propto)]$ according to (6). This vector verifies the presumed result.

In contrast to example A, project 2 additionally is dependent on project 1 in example B. In this case, one would expect a direct increase in importance of project 1 and an indirect increase in importance of project 5, as the status of project 1 increases and therefore contributes more to the status of project 5. The corresponding alpha centrality vector $\boldsymbol{x}^{\boldsymbol{T}} = [(1{+}\propto), 1, 1, 1, (1 + 4 \propto + \propto^2)]$ is in line with the expectations.
While examples A and B implicitly assume equal intensities of the existing dependencies, example 3 includes different intensities represented by weighted arcs. Representing the logical weighted extension of the one in example B, the alpha centrality vector of this example is $\boldsymbol{x}^{\boldsymbol{T}} = [(1 + 5 \propto), 1, 1, 1, (1 + 4 \propto + 5 \propto^2)]$.

3.5 Application Example

To demonstrate how this procedure can be used in practice, we illustrate the application with an example. Since we were not yet able to gather corresponding data, the intensities of project dependencies are assumed in this example. However, the other circumstances are based on a real world observations. In our case, the company incorporates an in-house IT provider that recently changed its software development process from the waterfall model to a release-oriented model. As a result, its current IT portfolio includes some projects that actually are sub-projects of an ensemble, which due to innovation pressure has been subdivided into several standalone projects. Hence, the portfolio features a high level of dependencies, and overall includes 15 projects, ranging from small infrastructure to big software development projects, all of which must be implemented within the next five years. In this context, the company faces the question of how to allocate its limited resources in order to accomplish the portfolio on time and under budget. Therefore, a risk analysis shall be conducted in order to identify the projects most critical to the IT portfolio, due to its inherent dependencies. To do so, the company first needs to identify and quantify the dependencies between the projects; this is usually accomplished based on interviews with the IT portfolio manager and other experts from the project management office (PMO). In this example, the resulting values have been normalized to range from 0 and 1, and the corresponding dependency structure of the portfolio is shown in Figure 2.

Based on this dependency structure, we derived an adjacency matrix $\boldsymbol{A}$ denoting whether and to which extent the projects are related to each other. Each element $a_{ij} > 0$ of this $n \times n$ matrix indicates that project i is dependent on project j. For the calculation of an alpha centrality value based on (6), we assumed the vector $\boldsymbol{e}$ to be a vector of ones. Since we rather wanted to examine the criticality of projects based on their dependency structure than on their exogenous status, the scalar $\propto$ has been set to its upper boundary value $\frac{1}{\lambda_{max}(\boldsymbol{A})}$, with $\lambda_{max}(\boldsymbol{A}) = 0{,}8243$. Based on (6) we were able to derive criticality values for each project, listed in Table 2.

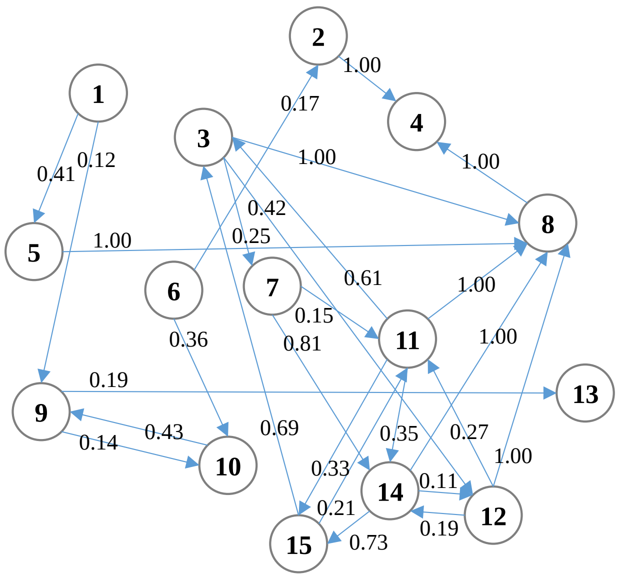

Fig. 2. – Example: Dependency structure

Table 2. – Example results

Nr.	$\frac{1}{\lambda_{max}(\boldsymbol{A})}$
1	0.0000
2	0.0000
3	-1.7551
4	-5.0874
5	0.0000
6	0.0000
7	0.0000
8	-4.1937
9	0.0000
10	0.0000
11	-1.1413
12	-0.9893
13	0.0000
14	-0.7126
15	-1.0879

Although this is just a very simple example, it illustrates the importance of dependency assessment quite well, as e.g. an allocation of resources based solely on project volume, as well as an inaccurate assessment of dependencies, would probably have led to a failure with regard to time or budget of the IT portfolio.

4 Summary, Conclusion and Limitations

The increasing demand for continuous innovations forces companies all over the world to assemble IT portfolios containing a high level of dependencies, while lacking appropriate methods to manage these dependence structures, as traditional methods rather focus on of cost and benefits than on the accurate assessment of direct and indirect dependencies. In order to empower companies to cope with the challenging task of successfully managing their IT portfolios, we explicitly focus on the assessment of the inherent dependency structure and derived a new procedure to assess the criticality of projects based on their dependencies. We therefore consider IT portfolios as IT project networks and draw on graph theory, as it is an approved means for the assessment of dependencies in network alike structures. In particular, we illustrate specific characteristics of IT project networks and evaluate different centrality measures regarding their appropriate applicability in this context. In doing so, alpha centrality was revealed as being a valuable approach in determining risk assessment of IT portfolios. It not only accounts for direct but also transitive dependencies, and shows that more critical projects contribute more strongly to the criticality of other projects they depend on. We depict the suitability of this measure based on its basic principles, and consequently propose an alpha centrality based assessment of dependencies to identify projects crucial to the success of the overall IT portfolio. To facilitate the comprehensibility and to verify the proposed procedure, we examine an exemplary IT project network based on its dependencies. The plausible results of the example application indicate that the proposed procedure is appropriate to analyze the dependencies between IT projects, and to assess their criticality to the overall portfolio's success. It furthermore highlights the practical implications of empowering companies to properly assess direct and indirect dependencies in their portfolio, as both common methods in practice as well as an inaccurate assessment of dependencies can lead to ill-considered decisions. Furthermore, the results especially emphasize that consideration of transitive dependencies is crucial for an appropriate risk analysis of IT project networks.

However, this approach is not without limitations and provides topics for further research. Referring to the three-stage cycle for research activities of MEREDITH ET AL. [15], we were not yet able to proceed from the explanation to the testing stage based on a real-world example, despite various efforts to gather data. We are currently in communication with a large IT consulting company in order to get data for the evaluation of a real-world example, which will lead to further research. Moreover, we do not explicitly consider different kinds of dependencies; however, this is considered acceptable as a first step, and the differentiation between various kinds of dependencies is a topic for a follow-up paper. Furthermore, this approach explicitly assumes that for a pairwise combination of IT projects, any kind of dependency can be quantified. Although, there are already some approaches quantifying different kinds of dependencies, further research should be encouraged to investigate appropriate measures in this respect. By assigning an initial status to each node of the network, independent of the networks dependency structure, this approach accounts for exogenous influences to the project's importance. Since determination of these exogenous influences is not in scope of this paper, they are assumed to be equally strong. However, exogenous influences, such as project budget or mandatory requirements, can determine a project's importance to different extents in the real world. Therefore, continuing research is required to include these kinds of influences in a comprehensive risk assessment of IT project networks.

Acknowledgements

This research was carried out in the context of the Project Group Business and Information Systems Engineering of the Fraunhofer Institute of Applied Information Technology FIT.

Grateful acknowledgement is due to the DFG (German Research Foundation) for their support of the project „Value-based Management of IT Projects“ (FR 2987/2-1; BU 809/13-1) making this paper possible.

References

1. Standish Group: Chaos Manifesto 2013. (2013)
2. Flyvbjerg, B., Budzier, A.: Why Your IT Project May Be Riskier Than You Think. Harvard Business Review 89, 23-25 (2011)
3. Radar Group: White Paper: The Impact of Data Silos in IT Planning. (2012)
4. Graves, S.B., Ringuest, J.L., Medaglia, A.L.: Models and methods for project selection: concepts from management science, finance and information technology. Springer, (2003)
5. Nguyen, B., Mutum, D.S.: A Review of Customer Relationship Management: Successes, Advances, Pitfalls and Futures. Business Process Management Journal 18, 400-419 (2012)
6. Buhl, H.U.: The Contribution of Business and Information Systems Engineering to the Early Recognition of and Avoidance of "Black Swans" in IT Projects. Business & Information Systems Engineering 4, 55-59 (2012)
7. Kundisch, D., Meier, C.: IT/IS Project Portfolio Selection in the Presence of Project Interactions – Review and Synthesis of the Literature. In: Wirtschaftinformatik Proceedings 2011. Paper 64. pp. 477-486. (2011)
8. Lee, J.W., Kim, S.H.: An integrated approach for interdependent information system project selection. International Journal of Project Management 19, 111-118 (2001)
9. Wehrmann, A., Heinrich, B., Seifert, F.: Quantitatives IT-Portfoliomanagement: Risiken von IT-Investitionen wertorientiert steuern. WIRTSCHAFTSINF 48, 234-245 (2006)
10. Kundisch, D., Meier, C.: A New Perspective on Resource Interactions in IT/IS Project Portfolio Selection. Proceedings of the 19th European Conference on Informations Systems (2011)
11. Beer, M., Fridgen, G., Müller, H., Wolf, T.: Benefits Quantification in IT Projects. In: Proceedings of the 11th Conference on Wirtschaftsinformatik (WI 2013), Vol. 1, 707-720 (2013)
12. Cho, W., Shaw, M.J.: Does IT Synergy Matter in IT Portfolio Selection? Proceedings of the 30th International Conference on Information Systems 160, (2009)
13. Zimmermann, S.: IT-Portfoliomanagement - Ein Konzept zur Bewertung und Gestaltung von IT. Informatik-Spektrum 31, 460-468 (2008)
14. Wasserman, S., Faust, K.: Social network analysis: Methods and applications. Cambridge university press, (1994)

15. Meredith, J.R., Raturi, A., Amoako-Gyampah, K., Kaplan, B.: Alternative Research Paradigms in Operations. Journal of Operations Management 8, 297-326 (1989)
16. Webster, J., Watson, R.T.: Analyzing the past to prepare for the future: Writing a literature review. Management Information Systems Quarterly 26, xiii-xxiii (2002)
17. Santhanam, R., Kyparisis, G.J.: A Decision Model for Interdependent Information System Project Selection. European Journal of Operational Research 89, 380-399 (1996)
18. Tillquist, J., King, J.L., Woo, C.: A Representational Scheme for Analyzing Information Technology and Organizational Dependency. MIS Quarterly 26, 91-118 (2002)
19. Zuluaga, A., Sefair, J.A., Medaglia, A.L.: Model for the Selection and Scheduling of Interdependent Projects. IEEE Systems and Information Engineering Design Symposium (2007)
20. Fischer, F.: Korrelationen von Risiken im Programm- und Projektportfoliomanagement: Ein hybrides Entscheidungsmodell zru Selektion alternativer Programme und Projektportfolien. Peter Lang, Frankfurt (2004)
21. Maheswari, J.U., Varghese, K.: Project Scheduling using Dependency Structure Matrix. International Journal of Project Management 23, 223-230 (2005)
22. Maheswari, J.U., Varghese, K., Sridharan, T.: Application of Dependency Structure Matrix for Activity Sequencing in Concurrent Engineering Projects. Journal of Construction Engineering and Management 132, 482-490 (2006)
23. Markowitz, H.M.: Portfolio Selection. The Journal of Finance 7, 77-91 (1952)
24. Butler, S., Chalasani, P., Jha, S., Raz, O., Shaw, M.: The Potential of Portfolio Analysis in Guiding Software Decisions. In: Proceedings of the First Workshop on Economics-Driven Software Engineering Research, EDSER-1, (1999)
25. Verhoef, C.: Quantitative IT portfolio management. Science of Computer Programming, 45, 1-96 (2002)
26. Zimmermann, S., Heinrich, B., Kundisch, D.: IT Projektportfolios mit Abhängigkeiten gestalten. (2012)
27. Dos Santos, B.L.: Justifying investments in new information technologies. Manag Inf Syst 7, 71-90 (1991)
28. Benaroch, M., Kauffman, R.: A Case for Using Real Options Pricing Analysis to Evaluate Information Technology Project Investments. Inform. Syst. Research 10, 70-86 (1999)
29. Taudes, A., Feuerstein, M., Mild, A.: Options Analysis of Software Platform Decisions: A Case Study. MIS Quarterly 24, 227-243 (2000)
30. Bardhan, I., Bagchi, S., Sougstad, R.: Prioritizing a Portfolio of Information Technology Investment Projects. Journal of Management Information Systems 21, 33-60 (2004)
31. Emery, D.R., Parr, P.C., Mokkelbost, P.B., Gandhi, D., Saunders, A.: An Investigation of Real Investment Decision Making with the Options Pricing Model. Journal of Business Finance & Accounting 5, 363-369 (1978)
32. Schwartz, E.S., Zozaya-Gorostiza, C.: Investment under Uncertainty in Information Technology: Acquisition and Development Projects. Management Science 49, 57-70 (2003)
33. Diepold, D., Ullrich, C., Wehrmann, A., Zimmermann, S.: A real options approach for valuating intertemporal interdependencies within a value-based IT portfolio management – a risk-return perspective. (2009)
34. Ullrich, C.: Valuation of IT Investments Using Real Options Theory. BISE 5, 331-341 (2013)
35. Carter, C.R., Ellram, L.M., Tate, W.: The Use of Social Network Analysis in Logistics Research. Journal of Business Logistics 28, 137-169 (2007)
36. Kim, Y., Choi, T.Y., Yan, T., Dooley, K.: Structural Investigation of Supply Networks: A Social Network Analysis Approach. Journal of Operations Management 29, 194-211 (2011)
37. Simon, D., Fischbach, K.: IT Landscape Management Using Network Analysis. (2013)

38. Newman, M.: Networks: an introduction. Oxford University Press, Inc., (2010)
39. Landherr, A., Friedl, B., Heidemann, J.: A Critical Review of Centrality Measures in Social Networks. Business & Information System Engineering 2, 371-385 (2010)
40. Sabidussi, G.: The Centrality Index of a Graph. Psychometrika 31, 581-603 (1966)
41. Freeman, L.C.: Centrality in Social Networks: Conceptual Clarification. Social Networks 1, 215-239 (1979)
42. Gould, R.V.: Measures of betweenness in non-symmetric networks. Social Networks 9, 277-282 (1987)
43. Bonacich, P., Lloyd, P.: Eigenvector-like measures of centrality for asymmetric relations. Social Networks 23, 191-201 (2001)