Criticality and propagation analysis of impacts between project deliverables

— Source link ⧉

Hadi Jaber, Hadi Jaber, Franck Marle, Ludovic-Alexandre Vidal ...+1 more authors

Institutions: American University of Beirut, Université Paris-Saclay

Related papers:

- Topology of large-scale engineering problem-solving networks

- Use of systems analysis and management systems in project planning and evaluation

- Modeling a Global Software Development Project as a Complex Socio-Technical System to Facilitate Risk Management and Improve the Project Structure

- New approaches for computer-based construction project planning

- Modeling and Simulation of Project Management through the PMBOK® Standard Using Complex Networks

Criticality and propagation analysis of impacts between project deliverables

Hadi Jaber, Franck Marle, Ludovic-Alexandre Vidal, Lionel Didiez

CrossMark

ORIGINAL PAPER

Criticality and propagation analysis of impacts between project deliverables

Hadi Jaber[1,2,3] · Franck Marle[2] · Ludovic-Alexandre Vidal[2] · Lionel Didiez[3]

Abstract The implementation of a management by deliverables and deadlines is based on detailed planning and strict control of deliverables. It is a strategic decision that reports to the project manager and a key element to the success of complex projects. Based on the modeling of the project elements and their interactions using weighted directed graphs, this article presents some contributions to anticipate potential behavior of the project. Topological and propagation analyses are made to detect and prioritize critical elements and critical interdependencies while enlarging the sense of the polysemous word "critical." We recommend a set of topological indicators suitable for project elements and interactions, which mainly allow us to discuss "How the impact of a project element affects other elements within the network? What is the collective influence of this element?". These indicators permit to prioritize project elements and their interactions by detecting the most influential ones taking into account the network structure. For instance, they permit to evaluate the collective criticality of project deliverables and to re-evaluate the priority of the risks associated with these deliverables by coupling the traditional features of individual risks with the topological indicators of the deliverables. Furthermore, some algorithms are applied to extract and visualize the propagation path between two elements within the network. For example, this allows to provide a vision of potential impact propagation between two project deliverables, either they are associated with two milestones or are critical. An application to automotive industry illustrates the benefits of the approach, and some perspectives are drawn for further work.

✉ Hadi Jaber
hadi.jaber@aub.edu.lb; hadi.jaber@centralesupelec.fr

Franck Marle
Franck.marle@centralesupelec.fr

Ludovic-Alexandre Vidal
Ludovic-alexandre.vidal@centralesupelec.fr

Lionel Didiez
lionel.didiez@renault.com

1   Department of Industrial Engineering and Management, Faculty of Engineering and Architecture, American University of Beirut, 1107 2020 Beirut, Lebanon

2   Laboratoire Genie Industriel, CentraleSupélec, Université Paris-Saclay, Grande Voie des Vignes, 92290 Chatenay-Malabry, France

3   Strategy of Quality Management Department, 1 Avenue du Golf, 78280 Guyancourt, France

1 Introduction

In a world of growing competition between firms, the time for the provision of new innovative products and services to market is becoming an essential part of the performance and success of an organization. The control of project delivery time requires expertise in strategy and organization, and the adoption of behaviors which permit a greater level of anticipation and stakeholder involvement. This must go far beyond simple mastery of planning and management techniques/tools.

The implementation of a management by deliverables and deadlines based on detailed planning and strict control of deliverables is a strategic decision that reports to the project manager. It is a key element of the success of complex projects. Management by deliverables is to find the match

between the needs of the project, the correct expression of these needs by appropriate specifications that pass through attentive listening to the customer, and a realization that meets the expressed needs. Describing the deliverables of the project in terms of precise specifications and requirements is an input to identify more accurately the work which will have to be done during the execution. The definition of reliable and stable requirements is an important success factor of projects (Yang et al. 2015).

Explicit descriptions of should-be and to-be activities and processes are key elements in implementing industrial projects (Romero et al. 2008). For instance, in the automotive industry, the purpose of project decomposition into documented deliverables is to provide a vehicle that meets the expectations of the customer. To achieve this, the company's actors are encouraged to imagine that vehicle, draw it, design it, simulate its production, start to offer the services, facilities, and equipment that will allow the customer to order it, and the company to manufacture and ship it. Project deliverables are then used during and after the project to simultaneously build the vehicle and the system to manufacture, transport, and sell it. This requires huge amount of information sharing between transmitters and receivers, across the company and its contractors, and across the project phases. This implies the necessity to correctly anticipate and keep under control the behavior of the project and its deliverables through time. The next section introduces the related work and the research question that will be addressed.

2 Related work

This section presents the classical project planning techniques to define and organize deliverables, the propagation phenomena between these deliverables, and the conceptual gaps about criticality analysis of project deliverables that implied to conduct this work.

2.1 Decomposing and organizing work

A complex project involves a myriad of activities which depend on each other in varied ways to accomplish project results (Browning 2013). Many methodologies do exist to define the specifications and requirements of a project. As underlined in (Cano and Lidón 2011), such specification definition process is the logical continuation of the stakeholders' expectations and constraints' identification. A proper and robust approach to identify requirements is all the more needed that the later a change of requirement occurs during a project, the more important its impact is, in terms of over cost, rework, etc. Some of these methodologies can be considered as "internal", meaning that the

deliverables of the project and their components are studied *a priori*, so that their specifications are correctly defined. Functional needs and solutions analysis is one of these methodologies. It permits to define the specifications of a system by studying its interactions with its environment in all the phases of its lifecycle (Yannou 1998). Other methodologies are, on the other hand, considered as "external", meaning that the requirements are defined without studying the deliverables themselves, but asking clients and stakeholders how they would specify them, such as customer listening methods (Garver 2003), (Gannon-Leary and Mccarthy 2010). As a whole, the conjoint use of such internal and external methods provides the best results in practice.

A project consists of deliverables that meet objectives that are realized through activities. These deliverables are themselves broken down into sub-deliverables and activities. Deliverables are the 'end products of a project or the measurable results of intermediate activities within the project organizations' (Association of Project Management 2000). The use of the term 'deliverable' views the main outputs of the project management system as being the new products, services or changed state that the project was set up to deliver (Bryde and Joby 2007). We consider that a deliverable is a term used in project management to describe a tangible object produced as a result of the project that is intended to be delivered to a customer (either internal or external). For example, requirements' specification and feasibility study are deliverables within a project. A deliverable could be a report, a document, a permit, or any other building block of an overall project. A deliverable may be composed of multiple smaller deliverables. It may be either an outcome to be achieved or a product to be provided (Browning and Ramasesh 2009). These deliverables are updated according to the changes and developments that occur throughout the project life cycle. They are archived at the end of the project and provide a practical basis for future projects within the company. For example, the final deliverables of product development projects in the automotive industry are documents for manufacturing vehicles in factories.

The construction of the schedule involves modeling graphically the dependency network between tasks. This is a structured decomposition of work. We must break the project into smaller subsets down (Lamers 2002). Many representations exist at the base of any planning construction. If the project is really a quasi-decomposable tree system, there must be a way to describe it as the interaction between subsystems is negligible compared to the interaction within each subsystem. Today, it does not exist. The decomposition is often done according to deliverables in a hierarchical tree called work breakdown structure (WBS), and the activity decomposition is finally displayed in a schedule called Gantt chart, but there are always numerous

interactions between these elements not displayable on conventional regimens (Miller 2009). Project scope and work planning includes the process of decomposing and organizing the entire project work into smaller units and thus more manageable packages of work (Tiner 1985). The traditional tool which permits to decompose and organize work in a project is the WBS. Such an organizational structure permits to manage more efficiently the execution of the project and measure its performance, given the fact that smaller units of work are in essence more easily accountable. Several rules should be kept in mind when the WBS of the project is built:

- The WBS is a deliverable-oriented grouping of project elements (Lister 2015).
- The WBS should be a bijection of the project scope: what is inside the WBS must be done during the project and what should be done during the project must be inside the WBS (Marle and Vidal 2016).
- Each parent unit of work, when decomposed into smaller units, should be preferably decomposed into 3–7 children. By doing so, the decomposition is useful and still easily understandable and manageable, the children units of work being sufficient enough to completely describe the parent unit of work (bijection) (Marle 2002).
- Each parent unit of work, when decomposed into smaller units, should be decomposed into homogeneous children units of work (for instance, according to time, geography, stakeholders, product components, etc). The mix of several natures of children for a same parent work is not desirable, for the confusion in scope delimitation and interaction management.
- Each elementary unit of work should be possibly measured in terms of cost, time, and performance (quality, project values, etc).

The WBS includes theoretically the project deliverables and its tangible results. Some mistakes come from approximations in the formulation and individual perceptions of the same formulation; thus, "design a car engine" can be interpreted by people as a goal: "the engine must be designed", as an activity "design a car engine", or as a deliverable "plans for the car engine". The formulation is not only interpretable for several actors, but also can change through time. The ideal WBS diagram recommended by PMI (PMI 2013) should only contain deliverables, objectives being apart in a separate tree, and even for activities.

2.2 Propagation phenomena between deliverables

Management by deliverables or control by results is a recent method of project management. This is an alternative to traditional project management techniques, historically resource-oriented or activity-oriented. For instance, the purpose of EVM methodology (Earned Value Management) is to compare the planned budget to the actual work performed (the earned value) and the actual cost to the planned budget. Fernandez stresses that "we cannot pilot the project by only following the schedule and budget. These are two fundamental concerns, but we should ensure the compliance of delivered features such as quality of technical implementation. Management by deliverables focuses on operational monitoring of the project; it focuses on results and allows the anticipation" (Fernandez 2011).

The managerial issues potentially associated with the mastering of impact propagation in a complex project are mainly related to its inability to be broken down into independent parts. This is true for all types of systems, whether natural, technical, or human. The consequence is that, whatever the way the system is broken down into, there will always be interdependencies between the parts. In a schedule, tasks are interconnected by dependency relationships. Each interaction can suddenly and strongly act on the impacted element, meaning that project behavior is subject to potential large and fast changes, sometimes called turbulent or chaotic phenomena. Therefore, it is not globally easy to understand and anticipate long-term behavior, mostly because of the number, diversity and uncertainty of interactions between project elements, including deliverables.

This implies risk of bad communication, bad coordination, or locally optimal decisions. Due to the number of interactions outside the official project structures, the danger is that the communication and coordination between actors may not be correctly done. Despite the events which disrupt the progress through the project, the propagation of impacts should be managed to ensure the continued achievement of targets in terms of quality, costs, lead time, product' technical performance, its industrialization and production volume, the image of the brand, and the associated partnership. The problems of impact propagation encountered in projects are usually due to inadequate anticipation. The aim is to master and anticipate potential cascading effects and their dynamics.

Some propagation of deliverable non-completeness errors can have significant consequences for the company, involving vehicles retouching: For example, door panels of a different color, or different shades of the shell under the steering wheel.

2.3 Gaps in criticality analysis of project elements

The estimation of task duration, and thus, the theoretical scheduling is uncertain. Some tools permit to cope with such uncertainty. For instance, advanced methodologies permit to determine the most likely critical path within a

probabilistic project network (Soroush 1994). Other models have been developed to propose solutions to the project scheduling problems with uncertain durations: based on sensitivity analyses (Samikoglu et al. 1998), Markov chain-based models (Hao et al. 2014), fuzzy logic (Shi and Blomquist 2012); (Masmoudi and Haït 2013), stochastic models, and associated heuristics (Bruni et al. 2011). The Critical Path Method is a mathematical analysis which identifies the sequence of activities that add up to the longest overall duration. In other words, this is the quickest way that the project can be done. Any delay affecting a task on the critical path is fully reflected in the project duration and, therefore, the end date. However, this analysis does not take into consideration the criticality of the deliverables. It only identifies the longest (duration) sequence of activities, regardless of their importance. We should not focus only on the critical path when we are evaluating which deliverables to monitor closely. In addition, we should focus on critical deliverables. A critical deliverable is a deliverable with a lot of risk, either because of its impact, its likelihood, or a combination of both. Some of these may or may not be on the critical path, yet they may be more important for staying on schedule.

Managing risk is not an isolated activity. It is a part of many project activities, including schedule management. There are other factors to consider when identifying items for which a timely delivery is critical. Existing modeling approaches have limitations when it comes to modeling the complexity of project deliverables. Hence, some propagation phenomena, such as chain reactions and loops, are not properly taken into account.

This article aims at analyzing and anticipating potential behavior of the network considering its structure and possible propagations, to help project managers making more reliable decisions. This is done by re-estimating project deliverable priority in terms of two complementary characteristics, their individual, and collective importance. The originality of this work is to combine both analyses and to propose a tailored set of indicators for collective importance assessment, based on numerous existing works applied on several types of systems and networks. These indicators have already been proven to be useful for such analysis in complex product development networks, like in (Braha and Bar-Yam 2007), since they permit to focus on specific nodes or edges of the network. Those may have a stronger influence on the global network behavior, since they are highly central in the network (acting like hubs). Moreover, other nodes or edges may have indirect consequences which are far higher than direct consequences, due to reaction chains or propagation phenomena.

Finally, we underline the following research question: How can one monitor and control the impact propagation within complex project deliverable network and make decisions to keep this network under control?

3 Acting on deliverables proactively considering their influence in the project network behavior

This section describes the process used to prioritize actions to mitigate complexity-induced risks throughout the deliverable network. This requires identifying possible dependencies between deliverables (paragraph 3.1); using network theory-based indicators to assess collective importance of deliverables within the network (paragraph 3.2); using propagation analysis as a complementary approach to identify deliverables that may have important indirect consequences (paragraph 3.3); establishing priorities using two types of indicators, the individual, and the collective criticality (paragraph 3.4); and defining actions to secure critical deliverables (paragraph 3.5), within and through project phases (Fig. 1). Figure 1 represents the monitoring of project deliverables at each milestone: controlling cost and quality and creating an effective action plan for the following milestone.

3.1 Modeling the project deliverable network

Because overt failure requires multiple faults, there is no isolated 'cause' of an accident (Cook 1998). Project complexity has three main drivers: size, variety, and interdependence (Vidal and Marle 2008). To deal with finer modeling of project complexity, we need a mathematical structure which can deal with a huge number of elements, of interdependencies, and also various types of project elements and interactions. As seen in the previous section, links between project elements can have a direction: a change in the direction can inverse the meaning and the interpretation of the link. Consequently, an additional requirement to the mathematical structure which should be used for modeling is that it should take into account the direction of links and their weights. The existing and emergent theories of system modeling like hierarchical representation of complex systems are based on weighted directed graph (Gunawan 2009). Therefore, the most adaptable structure to respect these requirements is the weighted directed graph. This structure can be presented in a matrix format. Complexity management by matrix-based modeling has come a long way. Matrix has become a widely used modeling framework across many areas of research and practice. A whole community was developed around the research on the Design Structure Matrix (DSM) originated by (Browning 2001; Eppinger et al. 1994a, b; Steward 1981).

Fig. 1 Project deliverable monitoring: at each milestone, a quality check is made

The dependency structure modeling (DSM) approach has proven to be a practical tool for representing and analyzing relations and dependencies among system components. The DSM approach has several advantages, such as the calculations inherent to the matrix format to get the benefits of different types of analyses. It avoids issues associated with the visual display of complex networks, especially in the case of structures including lots of interactions and even loops (Steward 1981), (Eppinger et al. 1994a, b), (Eppinger and Browning 2012). It is a highly compact, easily scalable, and intuitively readable representation to navigate across dependencies between elements.

A DSM is a square matrix, representing interactions between its elements, with the rows and columns identically labeled and ordered, and where the off-diagonal elements indicate relationships between the elements. In this article, we use the following convention for DSM orientation: an element's inputs appear in its matrix row and its outputs appear in its column. A Domain Mapping Matrix (DMM) is rectangular matrix mapping elements of a certain domain to elements of another domain (Akao 1990; Danilovic and Browning 2007).

DSM was applied by many researchers to study the propagation of failures between physical product components (Tumer and Stone 2003), (Stone et al. 2005), (Clarkson et al. 2004), between decisions (Jaber et al. 2015), and between activities (Eppinger and Browning 2012; Yassine 2004). The main originality of this article that it deals with managing the deliverables shared between activities of a product development (PD) project. In particular, we are interested in understanding the impact of incomplete deliverables (e.g., missing or poor quality) from one activity on subsequent activities, and how do these impacts propagate throughout the entire project. For instance, if we have bad quality of input data for a simulation model, these will trigger errors in the output results of the simulation, these results will trouble the receiver, and so on. To capture the complexity of exchanged and shared deliverables within a PD project, we propose a modeling framework based on the Design Structure Matrix (DSM). Then, several analyses are applied to act on deliverables proactively considering their influence in the project network behavior.

Project actors should not focus solely on the critical path at the expense of non-critical activities. Some non-critical activities became critical, which are almost always a sign of poor management. For these reasons, we work on project deliverables which are different from project activities/ tasks. A project activity may produce one or more project deliverables. We define two types of criticality to detect the real critical deliverable and enhance the management by deliverable in PD. This permits to detect some deliverable-related risks that disrupt in achievement of project objectives and not detected by the classical techniques.

Our DSM in this research work is the **deliverable–deliverable** matrix, called DD. It represents the relationships between deliverables, on which several improvements and analyses will be applied to understand and control the project behavior, more precisely the impacts' propagation analysis between its deliverables.

Moreover, we define two matrices as DMMs: the first one is the **Activity$_{Transmitter}$–Deliverable** matrix, which is built by modeling affiliation relationships between activities (transmitters) and deliverables. The second one is the **Deliverable–Activity$_{Receiver}$** matrix, which is built by

modeling affiliation relationships between deliverables and activities (receivers). Both matrices can be obtained using an algorithm applied on project plans to extract global interactions data from local interactions data. DD can be obtained thanks to the following formula when we have the same list of activity receivers and activity transmitters in both matrices:

$$\textbf{DD} = \textbf{Deliverable Activity}_{\textbf{Receiver}} * \textbf{Activity}_{\textbf{Transmitter}} \textbf{Deliverable}.$$
(1)

How to obtain DD and associated analysis and interpretations is detailed in the case study of vehicle development projects (Sect. 4).

3.2 Using topological network theory-based indicators to highlight elements due to their position in the network

This paragraph presents a literature review on topological indicators, their applications, and interpretations. For instance, (Clarkson et al. 2004; Giffin et al. 2009; Maier et al. 2014; Pasqual and de Weck 2012) have provided a good number of metrics applicable for analyzing network properties. More specifically, Braha and Bar-Yam have introduced such concepts in the context of large-scale complex product development networks (Braha and Bar-Yam 2004). These indicators permit to prioritize project elements and their connections according to their importance

within the network (the most influential elements and interactions taking into account the entire pattern of the network). In this article, we propose a set of the most adaptable to project elements that mainly allow us to discuss "What is the impact of an element to other elements within the network? What is the collective influence of this element?" These indicators permit to prioritize project elements and their connections according to their importance within the network (the most influential elements and interactions taking into account the entire pattern of the network). Figure 2 shows an example of a small network of project element with illustration of topological indicators. The size of the node (and its color) is proportional to the centrality indicators detailed in the following paragraphs, and the darkest and the biggest node corresponding to the element has the highest value of centrality.

3.2.1 Degree centrality

Centrality is the relative importance of a node within a graph. There are various measures to determine this ranking, such as "betweenness", "closeness", and "degree" are all measures of centrality. This measure gives a rough indication of the social power of a node based on how well they "connect" the network. Centrality is the degree functions that allow determining nodes with a large number of connections. It is also defined as the relative importance of a node within a graph. A node's degree centrality can be
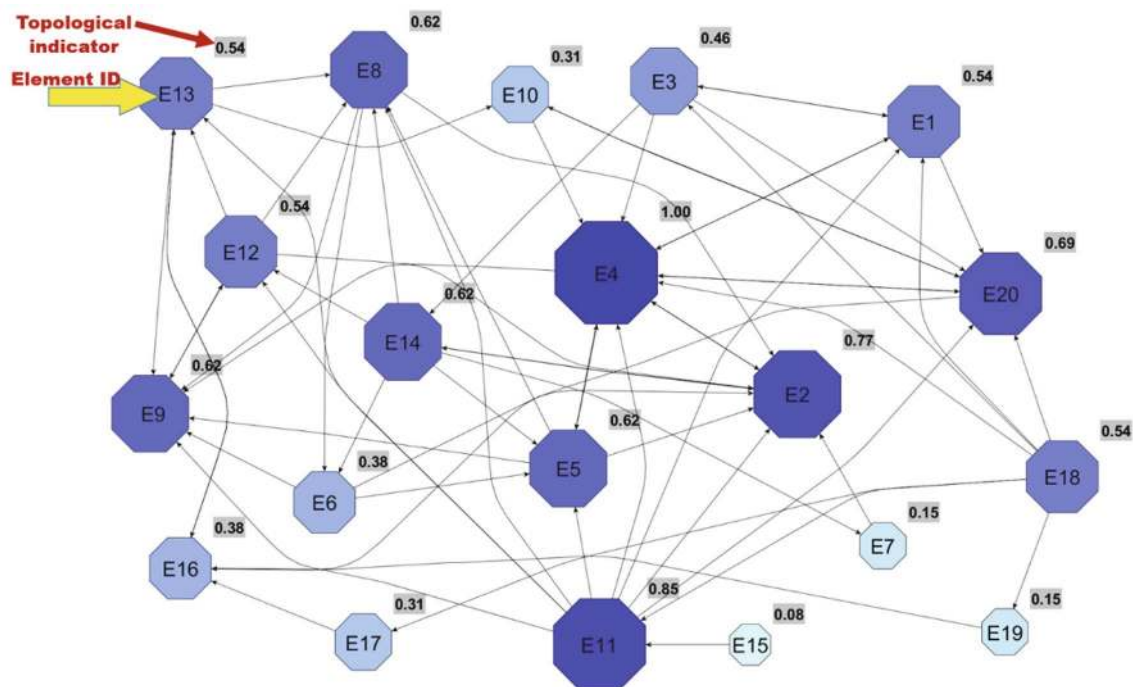

Fig. 2 Illustration of a network of project elements with topological indicators

defined as the number of nodes that are connected to that node in a graph. Freeman imposed categorized centrality measures into three basic categories degree, closeness, and betweenness along with the eigenvector-based measure proposed by Bonacich (1972) and Freeman (1977).

3.2.2 Betweenness centrality

Betweenness centrality denotes the number of pairs of nodes they lie between, or the number of paths that contain them (Freeman 1977; Guimera and Amaral 2004). It serves as an assistance to identify hubs in the network, particular nodes or interactions, which play the role of key passages for potential propagation. It is defined as the fraction of all the shortest paths in the network that contains a given node. In other words, is the sum of the fraction of all-pairs shortest paths that pass through a given node? Nodes with high values of betweenness centrality participate in a large number of the shortest paths. Betweenness centrality measures were applied in high-power grid selection and demonstrate very useful results in biological networks, road networks, web crawler, etc.

3.2.3 Closeness centrality

Closeness is based on the length of the average shortest path from one node to another. It focuses on how close a node is to all the other nodes in a network. It also describes the extent of influence of a node on the network. The degree a node is near all other nodes in a network (directly or indirectly). It reflects the ability to access information through the "grapevine" of network members. Thus, closeness is the inverse of the sum of the shortest distances between each individual and every other person in the network. The shortest path may also be known as the "geodesic distance".

3.2.4 Eigenvector centrality

According to eigenstructure analysis, the importance of a node is proportional to the importance of its connected nodes. Once again, such indicators permit to confirm previous results or to highlight surprising elements, elements that had not been seen as important, either by individual importance or by other topological indicators. Eigenvector centrality is a measure of the importance of a node in a network (Katz 1953; Bonacich 1972; Page et al. 1999). The idea is that even if a node influences directly only one other node, which subsequently influences many other nodes, then the first node in that chain is highly influential (Borgatti 2005). It assigns scores to the nodes based on the three following principles: (1) connections to more nodes contribute to the score; (2) connections to important nodes

contribute to the score; and (3) strong connections contribute to the score (Fang and Marle 2012), (Spizzirri 2011). This measure is used by sociologists to measure connection between players in social groups, and is implemented in Google's page rank (the system by which the search engine ranks the pages in its search results).

3.2.5 Core/periphery centrality

The centrality concept permits to examine the core/periphery structure of a network. The mixture of these concepts is the notion of a core/periphery structure, which is simultaneously a model of graph structure and a generalized measure of centrality. Here, all nodes can be regarded as belonging to a single group, either as core members or peripheral members. A common characteristic of core/periphery structures is that they have fairly short trail distances between pairs of nodes, which enable information to flow rapidly (Borgatti et al. 2013).

3.2.6 Direct and indirect reachability indicators (Marle and Vidal 2016)

Properties of a network can be highlighted by reachability indicators. The degree of nodes provides information on the local potential connectivity of a node X (Kreimeyer 2009). The number of outgoing/incoming edges is called the activity/passivity degree of a node.

The formulation of the calculation of the number of outgoing/incoming edges will be performed with the generic element X. This structure will enable easier transposition to other contexts, with X possibly being Actors, Product Components, Product Functions, or Project Deliverable in our study. However, X will be equal to Deliverable D in our study. NX is the number of elements {Xj}. XX is an NX × NX matrix with its elements $XX_{j1,j2}$ ($1 \leq j1, j2 \leq NX$) representing the interaction between elements $Xj1$ and X_{j2}.

The activity degree (number of outgoing edges) of a node i is calculated using the following equation:

$$\mathbf{AD}_i = \sum_j XX_{ij}. \tag{2}$$

The passivity degree (number of incoming edges) of a node *i* is calculated using the following equation:

$$\mathbf{PD}_i = \sum_j XX_{ji}. \tag{3}$$

The reachability matrix (RM) is built using the Floyd's sequential shortest path iterative algorithm, with $\mathbf{RM}_{ij} = 1$ if there exists at least one path from X_i to X_j (Floyd 1962). This reachability parameter has been used in several studies in the field of product development and project organization analysis (Feng et al. 2010; Braha and Bar-Yam 2004). The

powers of the adjacency matrix give information about potential paths of different lengths and about potential loops in the network (Warfield 1973; West 2001). The number of reachable nodes for a given X_i, called NRN_i, indicates the number of other nodes that Xi can impact directly and indirectly:

$$NRN_i = \sum_j RM_{ij}. \tag{4}$$

Similarly, the number of possible sources for X_i, called NPS_i, counts the other nodes that are connected or potentially connected to X_i:

$$NPS_i = \sum_j RM_{ji}. \tag{5}$$

These indicators on direct and indirect reachability degrees help to understand the global potential causes and effects of a node. The gap between the local potential impact and the global potential impact of a node expresses the potential events that might not be detected with classical direct cause–effect analysis. The existence of a potential path between nodes is useful for potential undesired reaction chain detection, even without any information about either the likelihood of the occurrence of the path, or its impact. Reachability degree helps us to understand the global consequences and sources of a risk of propagation, and enables us to classify them into different categories. Finally, the degree of nodes provides an indication of the local connectivity characteristics of the risk (Fang 2011). The number of reachable nodes indicates the number of other risks that a given risk can impact indirectly or directly. For arcs, the number of outgoing arcs signifies the activity degree of a risk and the incoming arcs give the passivity degree of the risk (Fang 2011).

We define the reachability index R of a node i as the summation of the number of potential successors (direct and indirect) and the number of potential predecessors (direct and indirect):

$$R(i) = NRN_i + NPS_i. \tag{6}$$

The normalized reachability index will be used in the calculation of collective criticality of project deliverables.

Moreover, the reachability matrix reports the number of existing path.

3.2.7 Definition of collective criticality index of a project deliverable

Let CCPD (n) be the collective criticality of a project deliverable "n". We define this collective criticality as the influence and importance of this deliverable in the network of project deliverables:

$$CCPD\,(n) = \frac{1}{5} * \Big[\textbf{Normalized Reachability Index}\,(n)$$
$$+ \textbf{Normalized Degree Centrality}\,(n)$$
$$+ \textbf{Normalized Betweenness Centrality}\,(n)$$
$$+ \textbf{Normalized Closeness Centrality}\,(n)$$
$$+ \textbf{Normalized Eigenvector Centrality}\,(n)\Big]. \tag{7}$$

If the centrality indicators cannot rank the project deliverables in some special cases of project networks, it is recommended to use only the reachability index to measure the collective criticality of project deliverables.

3.2.8 Group centrality

It generalizes the different centrality concepts from a single node to that of a group of nodes within the network. In addition, it is possible to evaluate the relative centrality of different teams or departments within an organization. Group centrality measure is a measure of the centrality of the whole group with respect to the individuals in the rest of the network, rather than to other groups. In group, centrality normalization is important, because different groups will have different sizes in the same network as compared to individual centrality, where normalization will be negligible (Everett and Borgatti 2012).

In new vehicle development projects, some activities are modeled to produce more than one deliverable at different dates. If we want to calculate the criticality of this activity, it is important to use the group centrality index of the produced deliverables.

3.3 Propagation behavior within the project deliverable network

In this paragraph, we propose an application of some algorithms to extract and visualize the propagation path between two elements within the network of project deliverables. For example, this allows to provide a vision of impact propagation between the project deliverables, with an option to focus on the chain that connects two deliverables associated with two milestones or on the chain that connects two critical deliverables.

In graph theory, a directed graph (or digraph) is a graph (that is a set of vertices connected by edges), where the edges have a direction associated with them. In formal terms, a directed graph is an ordered pair $G = (V, A)$, where V is a set whose elements are called vertices or nodes, A is a set of ordered pairs of vertices, called arrows, directed edges (sometimes simply edges with the

corresponding set named *E* instead of *A*), directed arcs, or directed lines.
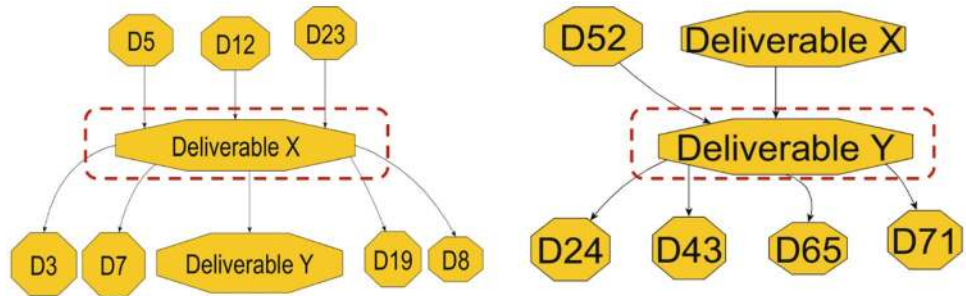
We propose three types of propagation-based analyses:

- a local, step-by-step web-like navigation without specific tools, but with a complete description of the direct environment of each element;
- an identification of the existence of potential paths between nodes and associated lengths;
- display the chains that relate two nodes.

3.3.1 Step-by-step propagation analysis

The first way to deal with potential propagation is to focus on a single element, showing all its interdependencies, but at a local level only. The idea is to give to the actor who will own or contribute to this central element the information about all its direct relationships. It is then possible to focus on one of these directly connected elements, which becomes the center of the diagram, and so on. This is similar to website navigation and enables direct and indirect relationships to be displayed on a user-friendly, complete (locally), and standard vision (Marle 2002).

For instance, Fig. 3 illustrates the case of complete representation of deliverables connected to **X**. One can see the classical interdependency of composition, its inputs, and its outputs. It is then possible to focus on **Y**. The right part of Fig. 3 shows that the sequential link between **X** and **X** is still displayed, but now, the rest of the information is about direct interdependencies with **Y**. Behind the deliverables, there are actors. This means that this navigation from deliverable to deliverable permits simultaneously to build communication paths between actors. This is illustrated in Fig. 3 for direct connections, but the principle is the same for longer chains.

3.3.2 An algorithm to identify the existence and the length of a potential path between two elements

We propose the use of a known algorithm in graph theory presented in Fig. 4 and which allows:

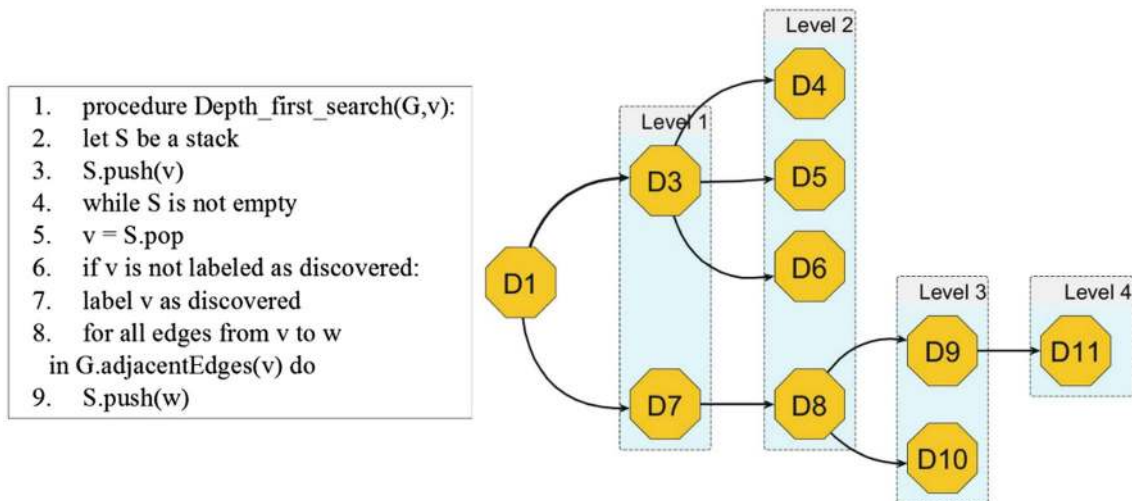- the identification of indirect consequences of an initial (un)desired event;

Fig. 3 Navigation from deliverable X-centered to Y-centered interdependency diagram

Fig. 4 Finding the direct and indirect neighbors for each deliverable in the project network

- the identification of indirect causes of a final (un)desired event;
- the detection of loops, which are characterized by the identification of a path which has the same initial and final nodes.

Figure 4 represents a depth-first search to find all nodes reachable from v. Given a digraph and source s, is there a directed path from s to v? If so, the following algorithm is designed to find such a path. Moreover, we can see in Fig. 4 an example of reporting these results for a source deliverable 'D1'.

This algorithm is implemented in Matlab, and the user interface is implemented in java. The developed tool permits to give in an ergonomic way, for each node, the connected neighbors with associated length of paths.

Display the chain that connects two deliverables:

In this analysis, we propose the use of the known Dijkstra algorithm (Dijkstra 1971) with an additional option that allows to remember the path which relates two nodes within the network, after that we report and visualize this propagation path between two elements (see Fig. 5). Extracting and reporting chains which relate deliverable sources and deliverable targets permit to create decision aid diagrams for project actors and thus reduce ambiguity and propagation of negative impacts between team members.

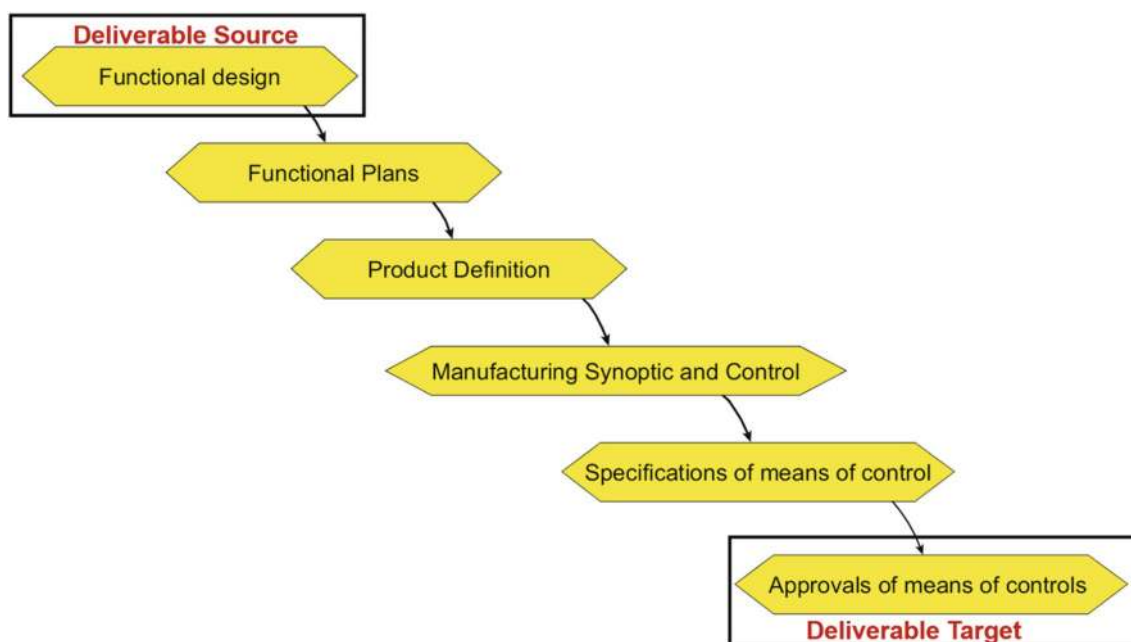The input of the algorithm in Fig. 6 is the matrix of the interactions between elements (the adjacency matrix of

Fig. 5 Displaying the path between source and target

```
1   function Dijkstra(Graph, source):
2   create vertex set
3   for each vertex v in Graph: // Initialization
4   dist[v] ← INFINITY          // Unknown distance from source to v
5   prev[v] ← UNDEFINED         // Previous node in optimal path from source
6   add v to Q                      // All nodes initially in Q (unvisited nodes)
7   dist[source] ← 0            // Distance from source to source
8   while Q is not empty:
9   u ← vertex in Q with min dist[u] // Node with the least distance will be selected first
10  remove u from Q
11  for each neighbor v of u:        // where v is still in Q
12  alt ← dist[u] + length(u, v)
13  if alt dist[v]:              // A shorter path to v has been found
14  dist[v] ← alt
15  prev[v] ← u
16  return dist[], prev[]
```

Fig. 6 Dijkstra algorithm which also returns the shortest paths (Dijkstra 1959)

weighted directed graph) and the source node. The output contains all distances and shortest paths from the node source. As a result, we can obtain the explicit shortest path between two nodes (source and target) in the network. For instance, applying Dijkstra algorithm to extract the shortest path between two deliverables improves the management by deliverables, by creating decision aid diagrams (see Fig. 5) for project actors, so they can anticipate the impacts propagation between these two deliverables.

In brief, we proposed a methodology of propagation analysis between elements strongly inter-linked and treat several cases and scenarios with an ergonomic and efficient way.

3.4 Prioritization using two types of criticality

As shown in Sect. 2.3, there is actually a lack of consensus on what deliverable criticality is. In this section, we propose a measure of deliverables criticality that take into account the individual and the collective criticality. Then, treatment strategies are provided to build a plan to act on deliverables considered as critical but for different reasons.

3.4.1 Individual criticality

The risk assessment of each deliverable should be made during the initial planning. It determines the probability that the deliverable of a task can be produced. There are three levels of risk to the project deliverables:

- The first level means that no high risk is linked to this task, similar deliverables were previously produced without particular problems.
- The second level means that unexpected difficulties may delay the delivery of the result. The exact result that the task is supposed to deliver has not been developed by the project team. It corresponds to the current state of knowledge, and the risk that the problem will not be solved is low.
- The third level means that it is not sure that the necessary knowledge will be produced with the resources allocated to the project. It is even possible that there are scientific or technical boundaries' conditions that prevent a positive response to the question about the task.

The allocation to these levels of risks can identify critical deliverables, the preparation of which is recommended as early as possible in the project. The temporal involvement may be less important if the deliverable can be reproduced in case of confirmed risk. In any case, the objective of risk levels is to anticipate optimally the possible dysfunctions to detect them as they emerge, communicate, and treat them spontaneously.

Fig. 7 Some causes of non-completeness of deliverables

Fig. 8 Some consequences of non-completeness of project deliverables

The quality of project deliverables encompasses four areas: correctness; timeliness; completeness; and flexibility of providing (Yang 2009). We define the concept of individual criticality of a deliverable as the assessment of its risk of bad quality (for example, non-completeness, see Fig. 7 that represent some causes of deliverables' non-completeness).

Figure 8 shows some consequences of non-completeness of project deliverables. The criticality level is normally divided into three degrees: simple, moderate, and complex (or respectively: green, orange, and red). We used a brainstorming to identify the individual critical features. Such features can be, for instance, the major deliverables to meet customer satisfaction; the deliverables associated with the critical path; and the identified late/non-complete deliverables from the return on experience of previous projects. Finally, the individual criticality is assessed using the following formula associated with the risk of deliverable non-completeness:

Criticality = Probability * Gravity * Detectability. (8)

The feedback of past projects revealed some examples of impacts' propagation about some purchasing choices of mono-sourcing, and some choices about the technical process of some pieces, that generate an impact propagation chain that amplify the charges and increase the logistics costs due to accumulated events. For instance, an error in input data of a simulation model will trigger bad quality of output results, which will bother the receiver of these results and so on. Then, we need to assess the collective criticality of project deliverables to decrease ambiguity, assist interface management and subsequently reduce risks of propagation.

3.4.2 Collective criticality of a deliverable

Many engineers interviewed within the automotive manufacturer Renault, cited several factors of collective criticality of project deliverables, such as: the deliverable is the result of many other deliverables and the deliverable is consumed by many receivers. Collective criticality analysis helps understanding the global importance of a deliverable, the global sources of impacts, and the global hubs influenced by many other deliverables that might not be detected with the classical direct cause–effect analysis (Marle and Vidal 2016). This analysis will show how one can deal with the difficulty to anticipate and control the consequences of complexity by proposing complementary complex-oriented mitigation actions. These actions may suggest to act on deliverables (e.g., to modify X to get X′), but sometimes on other elements or on other attributes than classical analysis output. Moreover, complementary indicators may involve different strategies like acting on an interaction (e.g., to get I′(X1, X2) less influent on the system behavior) or on an actor who manages an element (e.g., to assign a more appropriate A′ to X).

Figure 9 illustrates the additional information brought by the collective criticality analysis. The topological indicators represented in Sect. 3.2 permit to evaluate the collective criticality of project deliverables and to re-evaluate the priority of the project risks by coupling the traditional features of individual risks with the highest topological indicators of the risk network.

Once this two-axis information is available, some actions can be imagined and selected, and this is the object of the next paragraph.

3.5 Individual and collective criticality-based action plan

Basically in a network, it is possible to plan actions on nodes, on edges, or on chains (combination of several nodes and edges). It is to be reminded that here, nodes are deliverables, but of course, there are actors behind those deliverables, even if the latter are not directly modeled as the main focus object.

3.5.1 Acting on nodes

Over the course of a project (from its definition to its planning and execution), risks can occur and induce changes in the planning and, therefore, on the project time and cost (Marmier et al. 2014). To reduce the risk level in a project, it is necessary to define and apply a treatment strategy of the risk. The main idea is to combine several types of actions on specific nodes, these nodes being highlighted by classical or non-classical indicators (previous sections). Acting on project elements and their maturity consists in improving maturity to reduce the main internal weaknesses of the project (Gonzalez Ramirez 2009), but the basic short-term actions are to implement correctly what is

Fig. 9 Illustration of the additional information brought by the collective criticality analysis [adapted from (Marle and Vidal 2016)]
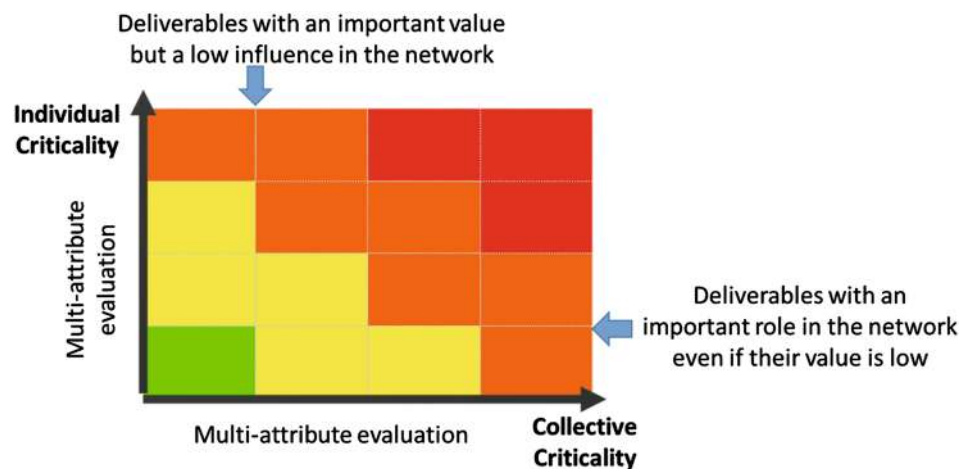
provided by the project office, or to simultaneously develop and implement something which was missing or immature. This gap between current and required maturity levels will have more or less consequences depending on the level of exposure to potential dangers. The more dangers there are, the higher the required maturity is.

3.5.2 Acting on edges and chains in the network

In classical methods, actions are decided on elements, such as, for instance, risks having the highest criticality or gravity. These actions correspond to the classical categories, which are avoidance, acceptance, mitigation, prevention, protection, etc. Based on refined evaluations and priorities, an updated response plan is developed, combining classical and innovative actions (Fig. 9).

Innovative actions include: (1) mitigation actions based on classical strategies but applied to new elements, depending on their refined values and rankings and (2) non-classical mitigation actions, which mitigate propagation occurrence instead of mitigating local problem occurrence. A complementary preventive action for accumulation or transition elements is to cutoff their input links or at least to reduce the transition probability values. Instead of acting on an element, the action focuses on its sources. Blocking the output links can be regarded as the action of confining the further propagation in the network. This is well adapted to source and transition elements. Instead of acting on the element, the action focuses on its consequences. This does
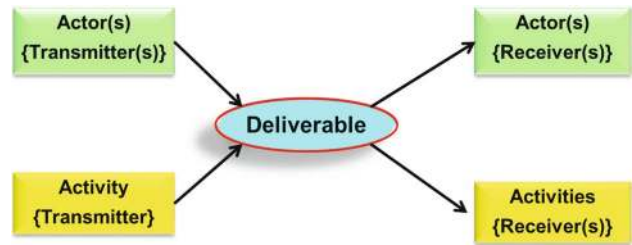

Fig. 10 Local data of interactions between elements of the development logic of new vehicles

not avoid the local problem, but its propagation and amplification to the rest of the project.

4 Application to vehicle development projects

Modeling, prototyping, and validating a new vehicle design entails dozens of subassemblies and hundreds of unique parts, all of which have complex engineering cross dependencies. Some design and engineering work can proceed in parallel; other tasks must be executed in sequence. These complexities must be modeled and factored into monthly, weekly, and daily planning buckets. At each milestone, a quality check is made. Between the milestones, the cost control monitoring is well organized at Renault, but the deliverable monitoring is partially made and not unified from one project to another.
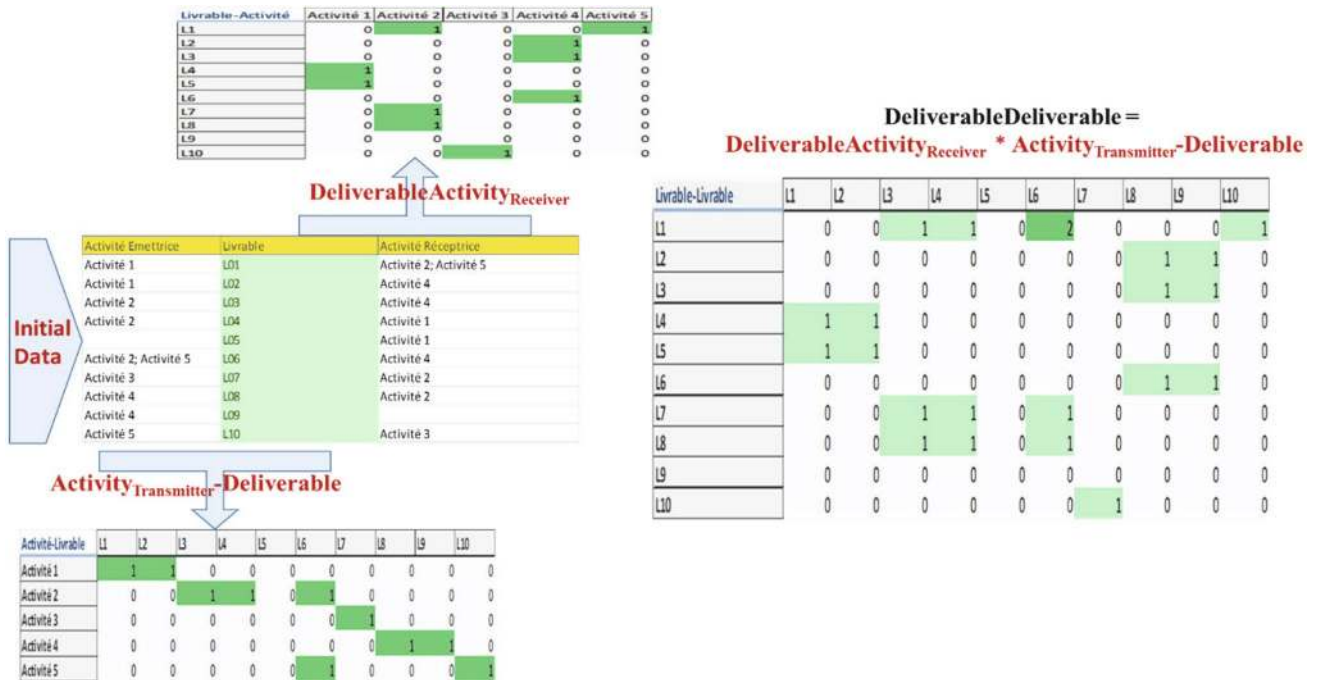

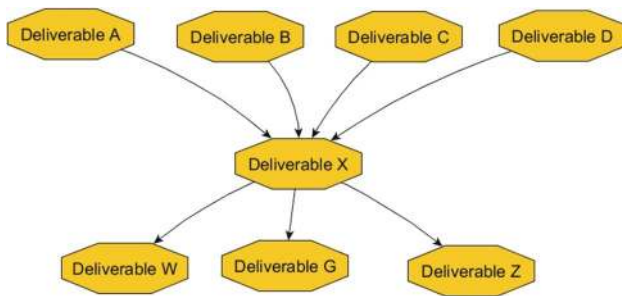**Fig. 11** Presumption of interactions between project deliverables

Fig. 12 Updated interdependency diagram centralized on deliverable X

Our aim is to develop action strategies to prevent critical deliverable-associated risks by providing decision support to improve the anticipation of shifting milestones. Even so, provide decision support for Quality-Assurance Engineers (who are responsible for milestones crossing agreements). The industrial need is to prioritize the most critical project deliverables.

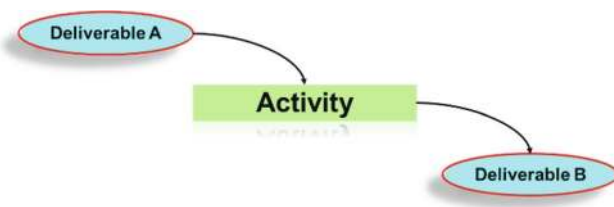### 4.1 Modeling the project deliverable network

Fig. 13 Presumption of dependencies between deliverables

The Renault Design System includes the development logic of new vehicles and associated processes, unifying processes, tools, and methods of vehicle engineering and mechanical engineering. Since 2010, the project steering within Renault follows a new development logic named V3P (Value up Product, Process, and Program). It includes activities to be undertaken by stakeholders and actors in the project to develop the mechanical parts and new vehicles. This new logic reduced the costs in projects around 30%, and improved the "Time To Market", between 4 and 6 months depending on the type of projects. Finally, it optimized the balance cost/value.

The entire company is organized around this logic. The timing and synchronization of the activities of all stakeholders must be respected for each phase. Each phase incorporates successive loops of convergence. Each loop aims a good result at the first attempt. The common references are shared before the loop start. Problems are treated

within each loop. The final milestone is a ratchet without turning back. We will employ the proposed modeling framework to analyze and improve the development logic of new vehicles, which its initial data are formalized locally as explained in Fig. 10. This is the input data to create networks of project elements. The first step of our analysis is the data collection about dependencies between project deliverables. Based on analysis of the processes' flow charts within the development logic of new vehicles (see Fig. 10), we made a presumption of interactions between deliverables by studying the paths of connections between deliverables via the activities' emitters and receivers (see Figs. 11, 12). We took into account during this assumption the time shift between emitting and receiving to delete fake links.

In addition, we have identified potential interactions between deliverables through the paths of connections via activities, as presented in Fig. 13. In addition, we detect and delete the false links related to temporal shift. This will be a basis to study the impact propagation between project deliverables.

For instance, the **Activity$_{Transmitter}$–Deliverable** matrix is built by modeling affiliation relationships between activities (transmitters) and deliverables. The **Deliverable–Activity$_{Receiver}$** matrix is built by modeling affiliation relationships between deliverables and activities (receivers). These matrices are defined as DMMs. Both matrices are obtained using the algorithm of global interactions data from local interactions data. The **Deliverable–Deliverable** Matrix, called DD, represents the relationships between deliverables, on which several improvements and analyses will be applied to understand and control the project behavior, more precisely the impact propagation analysis between its deliverables.

After presumption, we enriched our model of spreading impacts between deliverables by interviews with deliverables' owners and their emitting responsible. Then, we obtain a validate network of project deliverables. Figure 14 presents a zoom on small zone of this network, and shows the matrix of interactions between 254 deliverables. The total size of the verified matrix is about 2200*2200 deliverables.

4.2 Prioritizing the risks of non-completeness of deliverables with respect to their importance in terms of influence in the network

We defined a notion of deliverable individual criticality which is consistent and compatible with the different exploitations that are already made within Renault, for example: the frequency (percentage) of "green" validation of deliverables posterior to analyzing the feedbacks of 57 previous projects.

Fig. 14 Zoom on the interactions between 254 deliverables (verified dependencies)
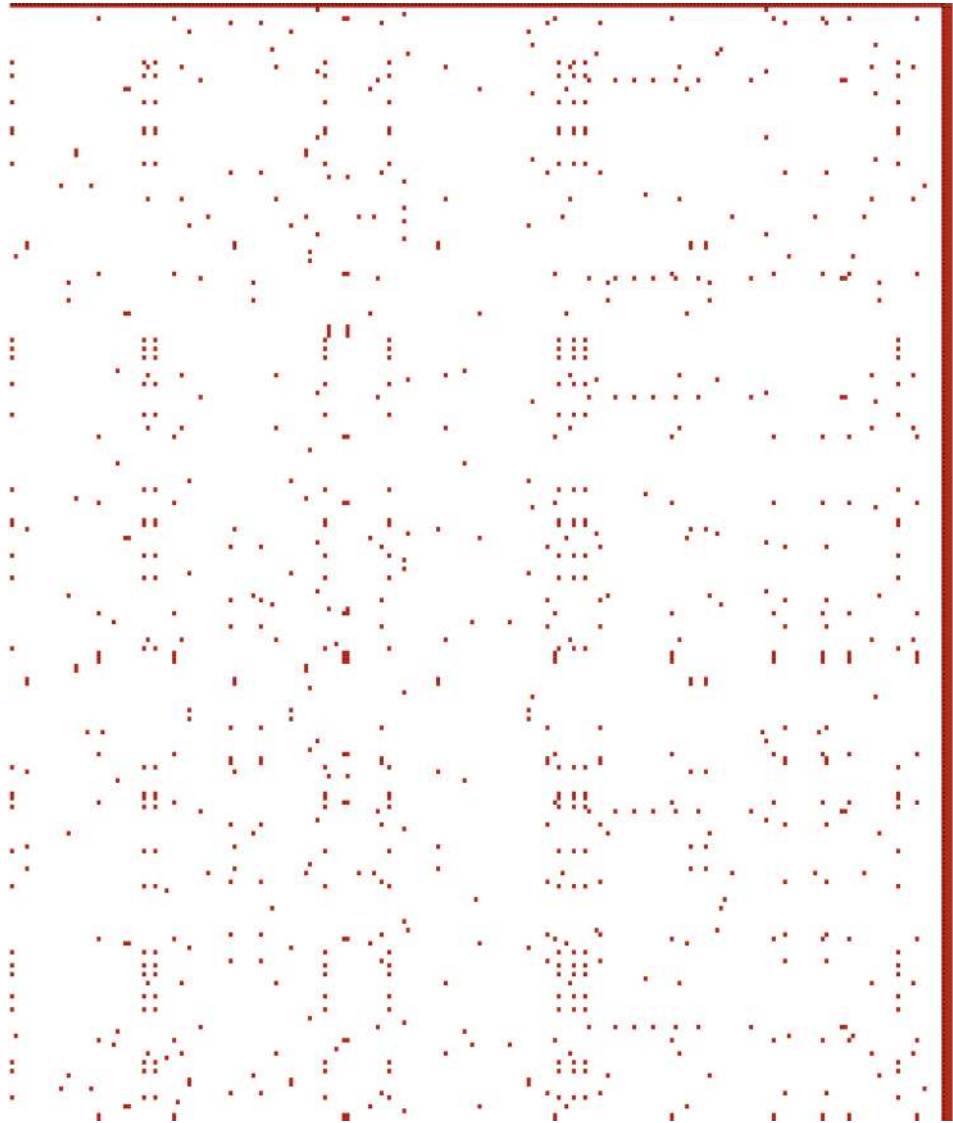
The assessments of this criticality were made by the risk assessment of deliverables' non-completeness. The collective criticality analysis is done through the identification and analysis of impact propagation channels in the network of deliverables as soon as possible in the project life cycle, in conjunction with the centrality indicators represented in Sect. 3.2. An example of the obtained results is given in Fig. 15.

Our assessment of a deliverable collective criticality is built from five indicators: degree centrality, betweenness centrality, closeness centrality, eigenvector centrality, and reachability index. Figure 15 shows these indicators and the aggregated collective importance. We combine four ways to detect different types of centrality. This means that there are more chances to detect central nodes. However, as mentioned in (Braha and Bar-Yam 2007), Fig. 15 illustrates that there are a few number of deliverables which dominate the

network in terms of centrality. Moreover, our assessment of collective criticality takes simultaneously into account centrality and reachability, meaning that in the case where large-scale projects would not include central elements, our assessment would still find some collective critical deliverables based upon the reachability index.

After assessing individual and collective criticality, we classify project deliverables in four main categories (see Fig. 16). The first one is for deliverables with a significant role in the network and important individual criticality value. This category includes, for instance, "Perceiving quality convergence" and "Product General Safety status". The second one is for deliverables with an important role in the network even if their value is low, this includes, for example: "PFE's checklist inputs". The third one is for deliverable with minor individual criticality value and without a significant role within the network. The fourth

Deliverable Name	Reachability Index	Degree Centrality	Betweennes Centrality	Eigenvector Centrality	Closeness Centrality	Collective Criticality	Individual Criticality
L91	100%	100%	100%	100%	100%	100%	71%
L96	100%	100%	100%	91%	100%	98%	20%
L56	91%	100%	100%	71%	94%	91%	28%
L52	93%	93%	50%	100%	100%	87%	5%
L82	91%	100%	78%	64%	88%	84%	10%
L54	93%	71%	82%	68%	93%	81%	82%
L64	90%	86%	60%	66%	91%	79%	69%
L77	97%	57%	98%	46%	87%	77%	32%
L45	93%	79%	55%	51%	93%	74%	100%
L75	76%	64%	55%	67%	92%	71%	3%
L58	86%	79%	53%	46%	83%	69%	44%
L99	65%	71%	21%	82%	97%	67%	38%
L98	87%	57%	44%	50%	83%	64%	77%
L74	67%	50%	15%	59%	90%	56%	80%
L69	55%	43%	17%	45%	81%	48%	19%
L57	68%	50%	19%	23%	76%	47%	49%
L71	44%	43%	16%	38%	82%	45%	45%
L92	50%	50%	17%	26%	76%	44%	65%
L90	51%	36%	6%	35%	81%	42%	71%
L49	53%	43%	8%	27%	74%	41%	75%
L83	64%	43%	12%	10%	66%	39%	28%
L51	49%	29%	8%	27%	76%	38%	68%
L93	50%	43%	21%	8%	64%	37%	66%
L97	38%	21%	4%	41%	78%	36%	16%
L84	39%	29%	6%	24%	76%	35%	12%
L88	47%	36%	8%	9%	65%	33%	50%
L87	42%	43%	7%	9%	63%	33%	96%
L70	43%	36%	4%	9%	64%	31%	34%
L79	36%	14%	7%	15%	69%	28%	59%
L44	32%	21%	4%	4%	57%	24%	22%

Fig. 15 Prioritizing the project critical deliverables

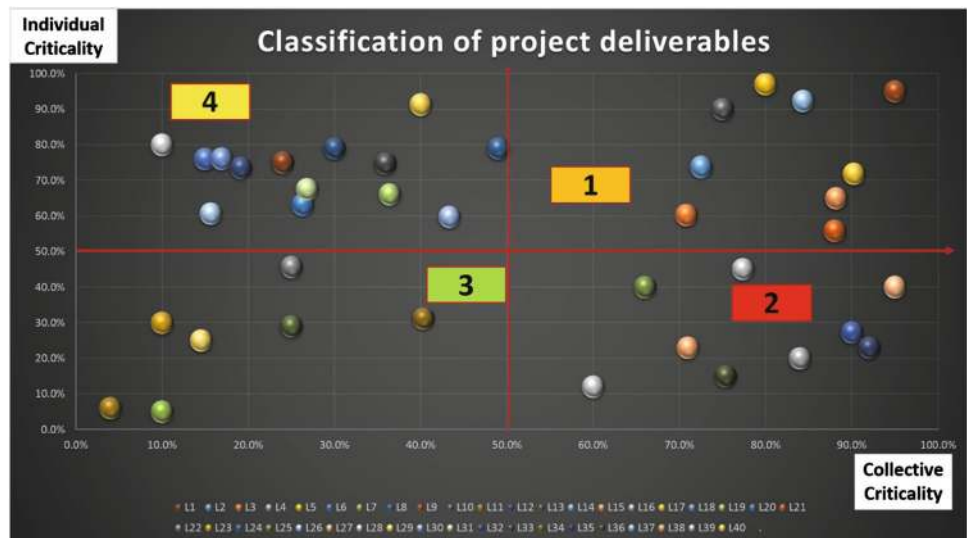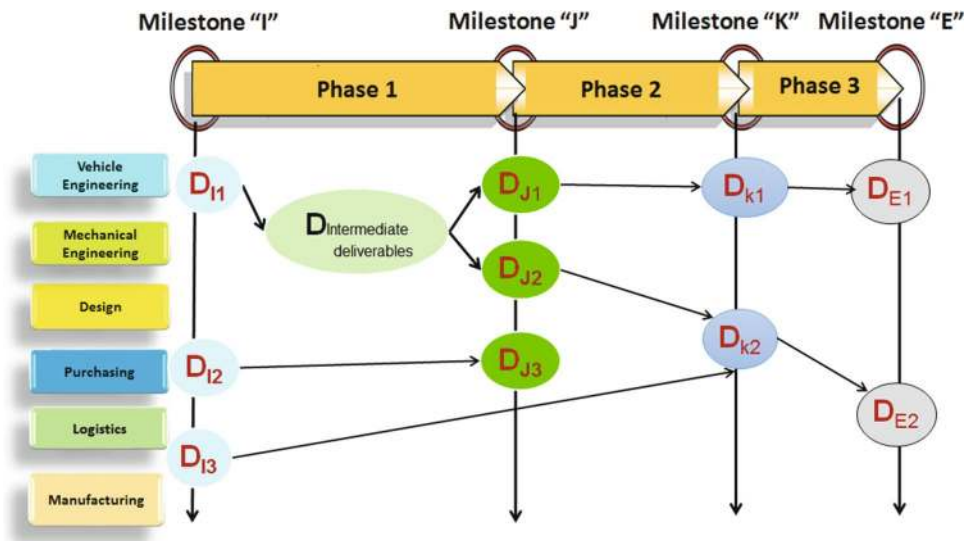**Fig. 16** Deliverables classification

Fig. 17 Impact propagation between project deliverables through milestones and organizational units

one is for deliverables with an important individual criticality value but a low influence within the network, such as "Project Team Training status".

Braha and Bar-Yam (2007) confirmed using both theoretical techniques and large-scale simulations that focusing engineering and management efforts on central elements of a network is likely to improve the performance of the overall network. Moreover, the failure of critical deliverables (individual or collective) is likely to affect the vulnerability of the overall product development process, which is the reason why this study has been done. This is illustrated in Fig. 16 where new deliverables are considered as important, for collective reasons, meaning that they are influent on the global network behavior. It is then important to take actions to change positively such deliverables, or to avoid actions which could be perturbations for these deliverables, as underlined by (Braha and Bar-Yam 2007).

The most significant improvement brought by our complementary analysis is the detection of deliverables in zone 2. This zone comprises potential deliverables that disrupt in achievement of project objectives but not detected by classical techniques. For instance, classical techniques consider many deliverables in zone 2 as non-critical, but in reality, a non-completeness of one of these deliverables can trigger a chain of impacts' propagation, and consequently, project delay, over cost, and sometimes, product bad quality. After detection of these deliverables, we can implement special controlling and monitoring of critical elements and making decisions to avoid the complexity-related risks. This means that we can reduce project complexity by acting either on nodes, edges, and chains within the network of project deliverables.

The proposed analysis gives additional information for the decision making in monitoring and controlling the impact propagation, since risks or deliverables may be considered influential for criticality and/or topological reasons.
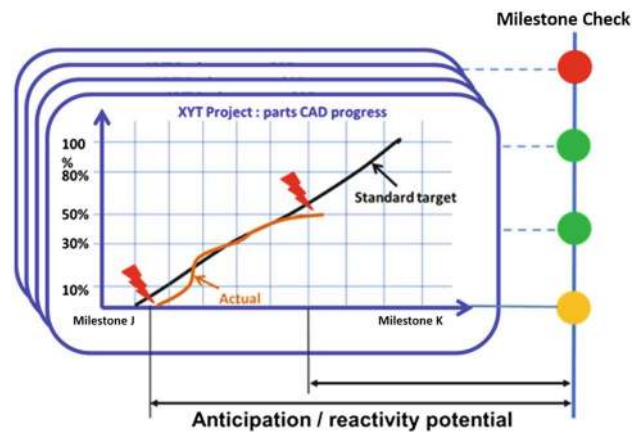

Fig. 18 Implementation of monitoring of project critical deliverables

That is to say, a deliverable taken individually may be non-critical, but through interactions could become the source of impact propagation to some critical ones. In the industrial application, after analyzing four thousand deliverables, we have detected many deliverable sources of impacts propagation within vehicle development project but individually considered as not critical. Renault implemented a rigorous monitoring process of these deliverables.

4.3 Results: monitoring of project critical deliverables

Finally, we provided an anticipatory vision of impact propagation between the deliverables, with an option to zoom in on the "chain" that connects two deliverables associated with different milestones (see Fig. 17) or the path between two critical deliverables. To do this, we used the proposed propagation analysis techniques in Sect. 3.3.

This helps on daily proactive management by deliverables. After the identification of critical deliverables and implementation of monitoring plans (see Fig. 18), we can get the right information to build the steering dashboard. To conclude, mastering the critical deliverables is based on:

- estimation of the remainder to make, and analysis of the deviations from the project target path (see Fig. 18);
- make the right decisions and manage corrective actions;
- ensure the quality of deliverables.

Risk management applied by the company integrates the anticipation of foreseeable risks and planning for possible solutions. However, the company does not control the management of unpredictable risks. Only the active planning with its decentralized and bilateral control system can compensate for this failure. Late deliverables continue to be a major impediment to project success (Barry et al. 2015). We should apply the prevention recommendations for the most common late deliverables (Top 100 selected) and lessons learned in managing late deliverables and mitigating their impacts. The implemented monitoring (see Fig. 18) improves cost, schedule, safety, quality, and organizational performance through a greater understanding of risks associated with late deliverables.

5 Conclusions

This paper contributes to engineering design research by introducing a methodology to monitor and control the impact propagation between project deliverables within complex projects and make decisions to keep propagation phenomena under control. This is done by modeling and analysis of project deliverable network to anticipate potential behavior of the project. Topological and propagation analyses have been proposed to detect and prioritize critical elements and interdependencies within the network. Individual and collective criticality indicators are proposed to enable an assessment of each project deliverable. It is thus possible to prioritize project deliverables and dependent deliverable chains. This allows a good highlighting of the critical deliverables and repercussion on the following deliverables that depend on them.

The main originality of this work is to combine two types of values in the criticality analysis, respectively, individual (based on classical criticality analysis using likelihood and impact), and collective (using the network analysis indicators). Priorities and actions are now based on a two-axis diagram instead of two separated analyses. The importance of these complementary analyses is huge, for two reasons. First, it is complementary and sometimes

contradictory with classical individual analyses. Second, it allows to detect potential disturbances that may be initially acceptable, but may propagate because of the number of elements they indirectly reach (topological analysis), or because of the amplifying behavior of the reaction chain they occur (propagation analysis) and, therefore, may these disturbances, and specific perturbations not be properly considered, then the consequences for the product development project are potentially huge, in terms of product performance but, moreover, on project performance (time to market, project cost, relationships with contractors etc). Second, the modeled element is specific, knowing that deliverables may be product- and project-related. A deliverable is modeled through its emitting and receiving activities and actors. The milestones determine time intervals, with the possibility to analyse propagation within these intervals, but also through different intervals. Indeed, milestones are more or less impervious, meaning that some undesired propagations can continue in the next development phases. Once again, a difference is made for decision-makers between one-phase and multi-phase propagations.

The industrial application on vehicle development projects is performed to build up and analyze the interactions-based project network. We are focusing on indirect risk analysis in vehicle projects via the analysis of propagation risks between deliverables, either on milestones or between two milestones. The obtained results demonstrate that the topological network analysis adds value to the classical project risk analysis, in identifying both the influential elements and the important interactions with respect to their role in the network behavior. Furthermore, the proposed propagation analysis gives additional information for the decision making in monitoring and controlling the impact propagation, since risks or deliverables may be considered influential for criticality and/or topological reasons. That is to say, a deliverable taken individually may be non-critical, but through interactions could become the source of impact propagation to some critical ones. The same analysis was done on the relationships between deliverables to evaluate the most crucial edges in the network structure. Overall, this reduces project complexity by mastering better the phenomenon of propagation. Based on the analysis outcomes, we demonstrate the effectiveness of using network theory for project element topological analysis. The proposed method is generic and could be applicable to a wide set of engineering projects for decision support.

The perspectives are: (1) to integrate the different complex network-based analyses that can be made on deliverables, but also on risks, actors, tasks, or product component architecture for instance; (2) to develop a more precise way to quantitatively assess the propagation of impacts through deliverable chains. Indeed, for the moment, the existence of such chains can be detected, the likelihood of triggering

of the event chain can be more or less precisely estimated, but it is very hard to estimate final impacts. This is mostly due to the fact that impacts are of diverse natures, and a consequence of an delay of a deliverable may be an over cost for another one, and then a requirement change for the latter one; (3) to combine such indicators with clustering techniques, to treat deliverable groups and not only individuals. This could give the opportunity to assess topological indicators at a group level, like in (Puzis et al. 2009). This could be done either by analyzing consequences of clustering on network structure and behavior, or using this structure analysis as targets for the clustering operation; (4) to visualize simultaneously the individual and the collective criticality on a single graph- or matrix-based representation; and (5) to consider the evolution of the network through time. For the moment, we chose two complementary ways to analyze complex networks: an analysis of the static snapshot, using network theory-based indicators, and an analysis anticipating the potential dynamical behavior of the network, using propagation tools. The network itself is dynamical, meaning that its nodes, edges, and values evolve through times (increasing/decreasing values, new/obsolete nodes/edges, etc). It is thus a perspective to simulate the evolution through time of the network, to update and re-anticipate dynamically its potential behavior.

References

Akao Y (1990) Quality function deployment. Productivity Press, Cambridge

Association of Project Management (2000) Syllabus for the APMP Examination (2nd Ed) Appendix 1 Glossary of Project Management Terms., Association of Project Management. ed. High Wycombe, UK

Barry W, Leite F, O'Brien WJ (2015) Late deliverable risk catalog: evaluating the impacts and risks of late deliverables to construction sites. J Constr Eng Manag 141:04014087. doi:10.1061/(ASCE)CO.1943-7862.0000950

Bonacich P (1972) Factoring and weighting approaches to status scores and clique identification. J Math Sociol 2:113–120

Borgatti SP (2005) Centrality and network flow. Social Networks 27:55–71

Borgatti SP, Everett MG, Johnson JC (2013) Analyzing social networks, 1st Edition. ed. SAGE Publications Limited

Braha D, Bar-Yam Y (2004) Information flow structure in large-scale product development organizational networks. J Inf Technol 19:244–253

Braha D, Bar-Yam Y (2007) The statistical mechanics of complex product development: empirical and analytical results. Manage Sci 53:1127–1145

Browning TR (2001) Applying the design structure matrix to system decomposition and integration problems: a review and new directions. IEEE Trans Eng Manage 48:292–306. doi:10.1109/17.946528

Browning TR (2014) Managing complex project process models with a process architecture framework. Int J Project Manage 32(2):229–241

Browning TR, Ramasesh RV (2009) A survey of activity network-based process models for managing product development projects. Production and operations. Management 16:217–240. doi:10.1111/j.1937-5956.2007.tb00177.x

Bruni ME (2011) A heuristic approach for resource constrained project scheduling with uncertain activity durations. Comput Oper Res 38:1305–1318

Bryde DJ, Joby R (2007) Product-based planning: the importance of project and project management deliverables in the management of clinical trials. R&D. Management 37:363–377

Cano JL, Lidón I (2011) Guided reflection on project definition. Int J Project Manag 29:525–536

Clarkson PJ, Simons C, Eckert C (2004) Predicting change propagation in complex design. J Mech Des 126:788

Cook RI (1998) How complex systems fail. Cognitive Technologies Laboratory. University of Chicago, Chicago

Danilovic M, Browning, TRTR (2007) Managing complex product development projects with Design structure matrices and domain mapping matrices. Int J Project Manage 25:300–314

Dijkstra EW (1959) A note on two problems in connexion with graphs. Numerische mathematik 1:269–271

Eppinger SD, Browning TR (2012) Design structure matrix methods and applications. MIT Press, Cambridge

Eppinger SD, Whitney DE, Smith RP, Gebala DA (1994a) A model-based method for organizing tasks in product development. Res Eng Design 6:1–13

Eppinger S, Whitney D, Smith R, Gebala D (1994b) A model-based method for organizing tasks in product development. Res Eng Design 6:1–13. doi:10.1007/BF01588087

Everett MG, Borgatti SP (2012) Categorical attribute based centrality: E–I and G–F centrality. Social Networks 34:562–569

Fang C (2011) Modeling and analyzing propagation behavior in complex risk network: a decision support system for project risk management

Fang C, Marle F (2012) A simulation-based risk network model for decision support in project risk management. Decision Support Syst 52:635–644

Feng W (2010) Dependency structure matrix modelling for stakeholder value networks. In: The 12th International DSM Conference. Cambridge

Fernandez A (2011) Les nouveaux tableaux de bords des managers, Editions d'organisation. ed

Floyd R (1962) Algorithm 97: shortest path. Commun ACM 5(6):345

Freeman L (1977) Set of measures of centrality based on betweenness. Sociometry 40:35–41

Gannon-Leary P, Mccarthy MD (2010) Customer care. Elsevier, ed. Philadelphia

Garver MS (2003) Best practices in identifying customer-driven improvement opportunities. Ind Mark Manage 32:455–466

Giffin M, de Weck O, Bounova G, Keller R, Eckert C, Clarkson PJ (2009) Change propagation analysis in complex technical systems. J Mech Des 131:081001. doi:10.1115/1.3149847

Gonzalez Ramirez N, 2009. Mesure de la maturité des projets: une approche pour améliorer le pilotage des projets automobiles. Ecole Centrale Paris

Guimera R, Amaral L (2004) Modeling the world-wide airport network. The European Physical Journal B-Condensed Matter and Complex Systems 38

Gunawan I (2009) Application of numerical design structure matrix method in engineering projects management. Operations and supply chain. Management 2(1):1–10

Hao X, Lin L, Gen M (2014) An effective multi-objective EDA for robust resource constrained project scheduling with uncertain durations. Procedia computer. Science 36:571–578

Jaber H, Marle F, Jankovic M (2015) Improving collaborative decision making in new product development projects using clustering algorithms. IEEE Trans Eng Manage 62:475–483. doi:10.1109/TEM.2015.2458332

Katz L (1953) A new status index derived from sociometric analysis. Psychometrika 18:39–43

Kreimeyer MF (2009) A structural measurement system for engineering design processes

Lamers M (2002) Do you manage a project, or what? A reply to "Do you manage work, deliverables or resources". Int J Project Manag April 2000 Int J Project Manag 20:325–329

Lister G (2015) Mastering project, program, and portfolio management: models for structuring and executing the project hierarchy, FT Press project management series. Pearson Education LTD, Upper Saddle River

Maier JF, Wynn DC, Biedermann W, Lindemann U, Clarkson PJ (2014) Simulating progressive iteration, rework and change propagation to prioritise design tasks. Res Eng Design 25:283–307. doi:10.1007/s00163-014-0174-8

Marle F (2002) Modèles d'information et méthodes pour aider à la prise de décision en management de projets. Ecole Centrale Paris

Marle F, Vidal L-A (2016) Managing complex, high risk projects. A Guide to Basic and Advanced Project Management. Springer-Verlag, ed. London

Marmier F, Cheikhrouhou N, Gourc D, 2014. Improvement of the planning reliability by the integration of human skills in project risk management, In: Logistics and Operations Management (GOL), 2014 International Conference on. IEEE, pp. 125–132

Masmoudi M, Haït A (2013) Project scheduling under uncertainty using fuzzy modelling and solving techniques. Eng Appl Artif Intell 26:135–149

Miller DP (2009) Building a project work breakdown structure visualizing objectives, deliverables, activities, and schedules, (ESI international project management series

Page L et al. (1999) The pagerank citation ranking: Bringing order to the web

Pasqual MC, de Weck OL (2012) Multilayer network model for analysis and management of change propagation. Res Eng Design 23:305–328. doi:10.1007/s00163-011-0125-6

PMI (2013) A guide to the project management body of knowledge: PMBOK Guide. Project Management Institute

Puzis R, Yagil D, Elovici Y, Braha D (2009) Collaborative attack on internet users' anonymity. Internet Res 19:60–77. doi:10.1108/10662240910927821

Romero F, Company P, Agost M-J, Vila C (2008) Activity modelling in a collaborative ceramic tile design chain: an enhanced IDEF0 approach. Res Eng Design 19:1–20. doi:10.1007/s00163-007-0040-z

Samikoglu et al (1998) Sensitivity analysis for project planning and scheduling under uncertain completions. Computers &amp. Chem Eng 22:871–874

Shi Q, Blomquist T (2012) A new approach for project scheduling using fuzzy dependency structure matrix. Int J Project Manage 34:503–510

Soroush HM (1994) The most critical path in a PERT network: a heuristic approach. Eur J Oper Res 78:93–105

Spizzirri L (2011) Justification and application of eigenvector centrality. Algebra in Geography: Eigenvectors of Network

Steward DV (1981) The design structure system: a method for managing the design of complex systems. IEEE Trans Eng Manag (3):71–74

Stone RB, Tumer IY, Stock ME (2005) Linking product functionality to historic failures to improve failure analysis in design. Res Eng Design 16:96–108. doi:10.1007/s00163-005-0005-z

Tumer IY, Stone RB (2003) Mapping function to failure mode during component development. Res Eng Design 14:25–33

Vidal L, Marle F (2008) Understanding project complexity: implications on project management. Kybernetes 37, 1094–1110. doi:10.1108/03684920810884928

Warfield J (1973) Binary matrices in system modeling. IEEE Trans Syst Man Cybern 3:441–449

West D (2001) Introduction to graph theory., Upper Saddle River. Prentice Hall, ed. NJ

Yang LR (2009) Impacts of automation technology on quality of project deliverables in the Taiwanese construction industry. Can J Civ Eng 36:402–414

Yannou B, 1998. Analyse fonctionnelle et analyse de la valeur. In: Conception de produits mécaniques, méthodes, modèles et outils., Tollenaere M, Hermes, eds. (ed)

Yassine A (2004) An introduction to modeling and analyzing complex product development processes using the design structure matrix (DSM) method. Urbana 51(9):1–17