Projects, *Super Sized Projects* and *Black Hole Projects*[1]

Charles Villanyi Bokor
The CERP Group
Charles_bokor@sympatico.ca

Aim of the Study (Issue): To advance the knowledge that Project Leaders need in order to be able to develop extremely large and complex projects and examine how the size of a project impacts the development methodology and the project governance needed to develop and deploy it. To explore the scalability of current methodologies, as well as the implications of ignoring project size when using development methodologies in their standard forms and by extension inspire future reflections to pro-actively prevent size induced project failures.

Lessons to Learn:

1. Projects have grown in size and the effort needed to develop them and have also become complex not just complicated;
2. The standard project development methodology is not scalable and has to be customized when used to develop enormously large projects.
3. BHPs are unique and current standard project management methodology and capabilities are inadequate to develop them and consequently BHPs are designed to fail.

Key Words: super-sized project, sub-project, enable, black hole project, project manager, project management, project leader, success criteria, paradigm shift, governance.

ABSTRACT

According to common belief, a project is a temporary endeavour. It has a defined problem to solve, a specific deliverable, a start date, a few stakeholders, an end-date, and an allocated budget that is based on the estimated schedule, the selected solution and the development team. This idea served its purpose in the '70s and '80s when application system specifications were written 'over coffee' on a napkin, when application system were coded in COBOL or FORTRAN, were developed on and ran on mainframe computers to automate specific operational procedures.

Some application system development projects today are very large, cost millions of dollars to develop, use many people with different skills and take years to deploy. Their primary focus is

[1] *Second Editions are previously published papers that have continued relevance in today's project management world, or which were originally published in conference proceedings or in a language other than English. Original publication acknowledged; authors retain copyright. This paper has been previously published in Software Quality Professional VOL. 17, NO 2/ March 2015; Business Systems Laboratory, International Symposia Proceedings January 2015; Requirements Network Group (RQNG), Nov 2012; and fmi*igf Journal, VOL 23, NO 1, 2011. It is republished here with the author's permission.*

not efficiency but to change a paradigm and enable the organization to reach its strategic objectives. Some of these very large projects evolve into larger than very large projects, which are different from the projects of old. These larger than very large projects have to be developed in a modified/customized development environment, using a customized project management methodology, and an adaptive and iterative systems design and development lifecycle (SD^2LC). They will cost much more than the originally allocated budget and/or take longer than the established schedule, will not meet all the usual or all the initial success criteria and will make the measuring of their success as challenging as developing the project itself. We will call these projects *Super Sized Projects* (SSP).

In the last two decades, the size of some SSPs exceeded all upper bounds. These projects have very large number of stakeholders, many different requirements and expected outcomes, unmanageably large development teams and a governance structure that becomes purely administrative. These innovative, enormous and larger than very large projects are conceived to impact the organization in such fundamental way(s) that most operational people reject them, even if they do not admit to their views. These totally overwhelming, all consuming, centers of the organization's focus are analogous to the most exciting concept in the universe and so we call this third type of project a *Black Hole Project* (BHP).

Developing *BHPs* in general, has proven to be beyond the capability of organizations using standard project management methodologies. They end up costing several times more than their poorly estimated and allocated initial budgets, last beyond their established development schedules, have on-going changes to their requirements, their design and the solution, as an integral part of the system design and development life cycle, and do not meet the success criteria that was set at the start. In short, using the present approach, BHPs are not viable projects, we do not know how to develop them and hence, they fail by design.

Projects, *Super-Sized Projects* and *Black Hole Projects*

Charles Villanyi Bokor

The size of IT development projects grew

According to a common belief, a project is a temporary endeavour undertaken to create a unique product, service or result and concludes when its specific objectives have been attained. It has a problem to solve, a start and an end date, a specific goal, allocated resources, and stakeholders and an expected result. It has an estimated budget and a schedule based on the selected business solution and the development team. Its charter describes its scope, its goals, its deliverables, risks, assumptions and sometimes the expected outcomes, based on how the project will create value for the corporation.

In the mid '70s, two Systems Analysts could write system specifications 'over coffee' on a napkin, undertake the analysis, feasibility, design-less development, testing and deploy a system in six months. The project would use one person-year of effort, as most of the projects at the time required single digit person-years of effort, and would have no risk analysis, change management, charter or governance committees. The application was usually coded by craftsmen in COBOL or PL/I, was submitted on punched cards behind the necessary JCL (Job Control Language) that everyone loathed and ran on the IBM 360 mainframe computer that everyone else used. Such an application system would use one set of input data, accessed a corporate database, solved one problem or automated a task that could be described in one sentence and produced a predictable output.

In the early '80s, a bank developed a national computer system. It linked its Canadian branches in each province using networked new-on-the-market 'mini' computers. The development project involved analysts, programmers, operating systems (OS) analysts, data base analysts and took a year and a half to develop and implement. It used 12 people and consumed some 18 person-years of effort. The vice president was the champion and provided oversight and the two stakeholders involved were accountable for the resources spent. It needed a large investment in new computers, transmission lines and new operating software. It had no risk management, change management or communications strategy and no business transformation. The project manager was accountable for attaining the stated output and business goal and was authorized to define the way to achieve it. He had no formal project management training, used no formal project management methodology, developed the work breakdown structure by hand and only evaluated rather than used, a new-on-the-market project management scheduling software. The project had an objective that could be described on less than a page and which it attained. Many other projects at the time were similar.

These projects were designed and developed to solve a specific complicated operating problem or automate a specific part of a solution that usually made the operation more efficient. They came very close to delivering their defined functionality and deploying as planned within their

fixed budgets, and defined schedules. Project development teams were generally less than ten people, development took less than 18 months, and total cost rarely exceeded $2.5 million. Few IT departments used a project management methodology, or a corporate standard to control development, but many had feasibility, functional and system design phases prior to the start of development. Admittedly, the number of cases where a 'Phase II' had to be defined, to deliver the functionality that was not done in Phase I, also grew over the years.

As the size of IT departments and application system development projects grew, federal government departments and private corporations using large development teams were deploying Enterprise Resource Planning (ERP) systems costing anywhere from five to over 20 million dollars. These larger projects used more resources, cost more, required more lines of code, started becoming complex not only complicated, and took a long time to be developed. As computer cycles became cheap and people became expensive, the project management profession evolved, methodology was formalized and projects kept on growing.

The Super Sized Project

In the late '90s the size of system development projects started to grow noticeably. "The bottom line is big systems have become too unwieldy." [1- McKendrick, 2010]. Big projects grew bigger, some became very big and complicated and some of these very big and complicated software engineering projects became bigger than very big. As an example, Linux OS version 7.1 released April 2001, contained over 30 million Lines of Code (LOC). Many of these bigger than very big projects rather than simply to deliver the project's outputs were developed to attain an expected outcome, an event, occurrence, condition or consequence that occurs as a direct/ultimate result of activities, in a cause and effect relationship. As such they were subtly different and had significantly different success criteria. Arguably, they required, not only benefited from a formal, somewhat modified project management methodology, a different emerging governance model based on allocated responsibility more than on entitlement, a modified development environment, being lead by a Project Manager (PM) with leadership abilities and visible executive management support.

As the solution of bigger than very big projects evolves during project development, the requirements change, and consequently the design also often needs to be changed meaning that the system design and development lifecycle (SD^2LC) needs to be iterative on very big projects. However, as many of the changes that customers request during development are not made, many of these projects will not meet all the success criteria but will cost more than the originally allocated budget, making measuring their success as challenging as their development and deployment. We will call these projects *Super Sized Projects* (SSP).

SSPs becomes the 'talk of the organization' projects that everyone wants to be closely associated with. They foster too much optimism about their impending successes, intentionally underestimating their challenges, including the concern for having selected resources, a modified methodology and visible organizational support. As the effectiveness of most of our standard project management methodologies used in standard development environments is inversely

proportional to the size of the project, SSPs deliver fewer than the expected outputs. Hence, developed as large, rather than *Super Sized* projects, SSPs fail more often than other projects. To successfully develop them, we need to make changes to:

- the *requirements change management* process. The proverbial Requirements Change Management (RCM) process is academically sound. However, does not readily allow the initial list of defined requirements, to be changed during the development process. It is usually more effective in keeping the project to its original budget and schedule, than in facilitating newly discovered changes that will aid the organization attain its outcome. This is mainly because the changes to requirements equates to changes to the system's design and an increase of the cost of the project due to new functionality or the rework of the already developed functions.

- the project's *governance structure*. For organizations to compete and excel in today's complex business environment, and develop SSPs, project management competence of the PM is still necessary but no longer sufficient. SSPs need project leaders who are authorized to make decisions, as better management of everything but the development team members, who need better leadership, is crucial [2- Janak, 2011]. If success is to be the only option, then SSP governance must assure that problems are resolved by experts and decisions are made, and that oversight committees are simply to oversee.

- the standard *development methodology* used: The 'water fall' SDLC and project development paradigm seldom work when an iterative and prototyping SD²LC is required. Wishing for a project development phase to be completed before the next phase is started is logical, common sense, predictive and as highly desirable as it is unrealistic when the project is enormous like an SSP. And while Continuous Risk Management, Communications / User Engagement Management is often simply 'lip service' implying that these processes exist, in fact often no one person or group is accountable for or manages the risks and communicates with or engages the team.

- the process used to *establish the budget*: It is hard, arguably, impossible to define and design an SSP's functionality before the solution that it is expected to facilitate or enable, becomes clear. Similarly, it is hard, arguably, impossible to readily elaborate an SSP's final solution before the system has been initiated and is under development. As the initial solution is vague, changes are continually asked for after project's initiation. Hence, the systems' design and functionality is never final, meaning that the established budget (and/or schedule) has to be expected to change within range. The range is determined by the minimum return on investment (ROI) the corporation is willing to accept, and the ROI that other projects could provide were the limited corporate funds spent on them. While the allocated budget increasing during the project's development is common practice, the total cost can also, decrease if the organization decides to only develop vital, i.e. about 20% of the identified functions, and not develop about 50% of the requested but not required functionality.

- The *development team composition*. It is hard to determine prior to system design and project initiation, the number of developers that will be needed to develop the project, when they will be needed, the skills that they will need to have and the level (junior, senior) of these skills. It is harder to get executives to assign the needed, i.e. accountable, dedicated and committed SMEs, rather than the available (cheaper) resources. Because an SSP project is not initially as well defined as a large project and as a result its scope and requirements change more frequently than that of 'projects of old', the required number, timing and skill levels of the needed people are even harder to predict. So, the SSP ends up with an inadequately large development team or with team members whose skills are inadequate or are not what is needed, except the members of the group that administers the project which does no development but grows in size. SSPs need a PM who is also a leader, to build teams based on peoples' gender, age, dressing habits, motivators and backgrounds, and is able, focus on creating value, and accept explicit accountability.

- The *concept of operations*. To produce the paradigm shifts in the organization's post-(system)deployment operation, the organization must not only be capable to deploy the complex software and technology needed by an SSP but also be capable to maintain it and engage people to use it. So, the changes to the standard operating processes and concepts have to be known, planned, and be part of a deployment strategy because people are reluctant to replace and forget the 'old ways'.

Research on this topic suggests that a set of (dependent) variables, necessarily more than one, are needed to define what constitutes a *Super Sized* system development effort. Further, that the relationship between these variables may be more important than the number of variables. While a specific point of demarcation that separates a very large project and an SSP is obviously impossible, we can define a very large project to be an SSP when a number of the following are true: it employs more than 25 people, last over 2.5 years, takes more than 60 person-years to develop, has over 100 K LOC [3- Gibbs, 2013], costs more than $8 million to develop and more than three times the development cost [4- Royce,1970], i.e. over $25 million, in total to deploy.

The Black Hole Project

If you think you know the answer to the problems facing BHPs, you are wrong.

Once projects started to grow they continued to grow. The evolving technology, lower prices and trust in the defined methodology used to develop projects spurred the imagination of organizations to reach for more elusive transformational goals. This lead them to conceive and undertake larger and larger SSPs to enable innovative and disruptive new business solutions. Based on over 20 years of experience, case studies and ongoing research, a few of the SSPs undertaken today exceed all reasonable upper bounds, and do not attempt to solve a problem but try to enable a vision (e.g. not only have a report on merchandise sold, but have real-time information on selling rate) or a new paradigm. However, defining an end state or outcome, does not equate to having a plan on how to get there, and designing the path to the outcome is often the problem. "Big projects fail at an astonishing rate." [5- Matta, Ashkenas, 2003] "Depending upon which academic study you read, the failure rate of

large projects is reported as being between 50%-80%. Because of the natural human tendency to hide bad news, the real statistic may be even higher. This is a catastrophe." [6- Dorsey, 2005] So, starting with the outcome, many such projects first design an initial business solution, lacking the knowledge of the final system that can enable it. We see numerous examples of descriptions of the expectation, the vision, goal or outcome, i.e. the end state, that does not describe the path or tasks to get there. These include 'Become smoke free for a healthy lifestyle.', 'deliver to users' expectations' or 'provide electronic health records for all Canadians' and 'provide service to process payments to 300,000 federal employees'. Another observation is that enormous SSPs last many years, have unmanageably large development teams that keep growing, and have a governance model that is composed of a very large number of overseers who have few explicitly accepted accountabilities and who form a group that becomes purely administrative.

A few of these SSPs are not only complicated, like a jet liner, but are also complex, like mayonnaise or a fertilized egg, so when their development is started no-one is able to explain entirely how they will enable the outcome (the larger the system, the harder to predict its behaviour - *General Systems Theory*). To allow clear deliverables or outputs, they get subdivided into multiple sub-projects, use significantly more than the required number of people, partly because of the unaccounted for overhead needed to coordinate the sub-projects; require continuous changes to the evolving design and scope; deliver twice as many sub-quality, only more or less integrated outputs than needed; and are developed slowly, in part because there are too many people and partly because many people leave them before they are finished.

Even if done right, such projects will not provide the expected value or facilitate the organization to easily achieve all of its stated goal. PMs of these sub-projects, even if qualified, and some due to the large number of PMs needed are not, are focused on being project managers. That does not equate to being focused on managing their projects. So, they manage their sub-project, follow standards and policy, develop the requested deliverables right, on time, on budget and report detailed achievements. They are focused on the importance of their sub-project, not on their sub-project's outputs integrating with other outputs or the outcome of the portfolio of sub-projects delivering value. They rely on the decisions of several committees, who traditionally defend the *status quo* that they are SMEs of, so even if a number of these sub-projects succeed, the portfolio or collection of sub-projects, at best, delivers an updated version of the capability that the organization already had with modules that more or less work together. As little effort is allocated to facilitating deployment and explicitly managing the resultant change, the goal to deliver a vision or the intended paradigm shift will not be attained and the project will in general be considered a failure.

These exceedingly very large SSPs will have numerous stakeholders, who have different needs and desired outcomes. These they will change as the development proceeds and they see a better solution, which in turn will require changes to the design, significantly impacting the development that is already underway. As system success is often measured by being 'on-time and on-budget', many of the requested changes that would require additional effort, will not be accepted (requirements change methodology not being outcome focused).

To exemplify the impact that changes to the requirements of exceedingly large SSPs can have, let us say that a mega SSP project has one change made to its requirement per month of its development. Let us say that each one of these monthly changes represents a one percent change (note: large software project requirements in general change and increase at the rate of 2% per month; "… the fact that requirements change at about 2% per calendar month …" [6.2- Jones, 2007]), increase or decrease, to the total design and development effort. Then, if the project was to take 100 person-years to develop, each one percent change would represent a one person-year of effort per month. As each formal change request can, and arguably does also imply a one percent increase in the analysis needed to determine the feasibility and the impact of the request and a one percent increase in the additional governance effort, remembering that these projects use large oversight committees in order to oversee and approve/disapprove changes, each design or functionality change request can become a three percent increase in the total effort needed to deliver the project [6.5- the Thomsett company 2000]. Over a one-year period, a three percent monthly change (3% compounded 12 times) can amount to a 40 % yearly increase in the required effort. Even if the requested change results in a one percent reduction of the original development effort, it still implies a one percent increase in the total effort expanded, as it still needs a one percent increase in the effort for the analysis and one for the governance. Hence if six requests are for increased functionality and six are for decreased functionality, the actual total work effort including analysis and governance required would be 23% more (represents a three percent increase six times and a one percent increase six times), even though the development effort could be claimed not to have changed.

Exceedingly large SSPs generally start consuming all of the available corporate resources and attention. If good progress is made, then they impact the organization in such fundamental way that most operational people reject them, and many managers whose jobs they impact resist the change. "When asked to select the top three reasons people resist change, 44 percent of employees say they don't understand the change they're being asked to make, and 38 percent say they don't agree with it." (7- Aguirre, von Post, Alpern,2013) If little progress is made, then the team further increases the effort and spends even more resources, due to an increased optimism for success. In addition, the first iteration of the deliverables generally shows marginal or no benefit or consequential value compared to normal projects or other more manageable SSPs. These totally overwhelming, all consuming, centers of the organization's focus are analogous to the most exciting concept in the universe and so we call this third type of project a *Black Hole Project* (BHP).

EXAMPLES: *Black Hole Projects*
In January 2001 **McDonald** conceived of a global, real-time digital network called *Innovate*. It was going to be the most expensive and extensive information technology project in the company's history. The project evolved and eventually was going to cost $1 billion. As McDonald's found out, their expertise in mass-producing cheeseburgers and French fries had little relevance to the world of software development, integration and implementation. The project was scrapped after $170 million was spent.

Apple Computer abandoned the development of the *Copland* an updated version of the Macintosh operating system after spending around $200M. Development began in 1994 and

officially ended in August 1996.

The **Canadian Firearms Registration** Project's cost to taxpayers, went from an original estimate of $20 million to more than $1 billion. Some Liberal MPs are alleging ministerial incompetence. "The government … plans to spend $50 to $60 million every year to operate the registry."

Canada Health Infoway has approved funding for 294 projects across Canada as of December 31, 2009, and some $1.6 billion in federal funding. Studies have shown the initiative will cost between $10-billion and $12-billion "...when all is said and done...".

Boeing's attempt to build the 787 Dreamliner, includes integrating some 380 major subcontractors. The resulting parts have been fitting together less than seamlessly.

NASA's James Webb telescope budgeted for $1.6 B ballooned to a cost of $8.7 B in one decade, rising $2.2 B in 2010.

Phoenix Pay System. The 'fixes' to the $315 million, 230 people project, were expected to make it work one year after it was implemented. After nine months of adjustments to its twice delayed, disastrous implementation, its overrun (in 2016) is beyond $56 million.

Shared Services Canada's email conversion, was a forgone conclusion.

A multi-department initiative to consolidate 1,500 different websites under the banner of **Canada.ca** started in 2013. Originally contracted out for $1.5 million, this **website project** could balloon and end up costing $1 billion (Anthony Furey, POSTMEDIA NETWORK, Dec 15, 2016). It was supposed to be finished by the end of 2016 but has been pushed back to the end of 2017. "…at this point (late 32016) they've migrated about 0.5% of the web pages," said Mike Gifford, the founder and president of Ottawa-based Open Concept Consulting.

As projects grow in size so do the number of requirements, the number of changes to the requirements and its cost.

As the above examples indicate, BHPs are enormous. They continually grow as the business solution, evolves at the same time and partly because of how the enabling system is being developed. Eventually, the system surpasses the capability of most system development experts to fathom them, the organization to develop them and to support the required technology, as the solution surpasses the organization's capability to integrate the novelty into their operations or to get value from them. Once started, BHPs involve so many high-profile people, have such high sunk-costs and have such great momentum as to be next to impossible to stop. They are much like a tsunami when meeting a milestone. While many managers take ownership of the vision that they are to achieve, no one takes accountability for resolving any of the many challenges, problems, or issues that these projects create. For example, the F-35 fighter jet, was mentioned in some "2,900 articles … the majority of which were critical" [8- THE OTTAWA CITIZEN, 2012]. "The planning of *Phoenix* and the broader pay transformation initiative was flawed. But casting blame doesn't help…" according to Judy M. Foote Minister of Public Services and Procurement Canada - Before the Standing Committee on Government Operations and Estimates OTTAWA, Ontario, November 29, 2016. A BHP, as expected, will have on-going changes to its solution,

design, functionality, and will not meet the success criteria that are set at the start and kept unchanged as it cannot be controlled by any current standard project management methodology. It requires an active and accountable Executive Management Steering Committee, a Project Leader, a customized project development methodology and a (project) governance model as much as it requires developers. BHPs are not projects, as per the criteria in the definition, and our records show we cannot develop and deploy them the same way we develop and deploy most projects. But because we think we can, and senior executives say we can, developers will not admit that we cannot and so they ignore all professionalism and design BHPs, to fail. "When we surveyed members of the Senior Executive Service, the elite ranks of federal managers, 60 percent said that government was less capable of executing large projects today than it was thirty years ago.". [10- OAG, 2011] Only the exceptional one is successful, often due to luck.

Moving Forward

It is even more daunting to define a BHP than to define an SSP, as a clear demarcation or algorithm that reflects the complex relationship amongst its variables, is even more elusive. However, a seasoned PM 'knows' one from the other. For everyone else, we can use the variables or quantifiers used to define an SSP, academic management and human behaviour theory, to explore the threshold that an SSP has to exceed in order to be classified as a BHP. Preliminary research done on this topic suggests that a BHP employs more than 250 people, takes more than 700 person-years to develop (only 200 person-years according to Trendowicz and Jeffrey) [11- Trendowicz, Jeffrey, 2014]; has over 20 M LOC ("… approximate range of 10 K function points in size. … roughly equal to about 1,250 K statements in the C programming language." [6.2- Jones, 2007]); takes more than 3 years (in accordance with Trendowicz and Jeffrey); costs more than $50 million to develop and more than three times the development cost, i.e. more than $150 million to deploy and elaborate some 400 lines of documentation for each LOC. In addition, a BHP has a significantly large number (> 20) of people on its governance team overseeing it as well as a large number of stakeholders. The following table quantifies the variables for closer inspection, allowing criticism of the concepts to surface.

	PROJECT	SSP	BHP
Number of developers	10†	> 25	> 250
Person-years of effort	7†	> 60	> 700
Size LOC	25 K†	> 100K	> 20 M
Cost to develop Cost to deploy	$750 K† < $2.5 M	> $8 M > 3X cost to develop	> $50 M > 3X cost to develop
Duration	1 year†	> 2.5 years	> 3 years
Governance (management) team	PM and appointment	Formal standardized	Many
Stakeholders	A few appointed	Formal standardized	Many

Table #1: Comparing Projects, SSPs and BHPs † Approximately

What the astute reader will notice is first, that different people have come up with different numbers for the variables above. This and the lack of precisely established relationship between the variables is not surprising. Most notably we have to mention the work of Capers Jones, Boehm [12- Boehm, 1981], and Trendowicz and Jeffrey [11- Trendowicz, Jeffrey, 2014, pg51]. Secondly, the astute reader will notice that we also have so far avoided defining the formula or algorithm that expresses how these variables relate. This will be left for another discussion. But what the variables indicate by themselves is that a BHP is not only an enormous project but also a different kind of a project. It is not one to solve a problem, or improve efficiency but to enable an outcome. It is to gain competitive advantage for the corporation, better access to information and decision making. BHPs are unique, conceived to change a paradigm, and they cannot be developed using the standard project management paradigm, methodology and project management structure and in general are unsuccessful due to the unpredictable interplay of several variables and conditions. BHPs are complex and no simple solution, no matter who pretends to have one, can be their answer. The reasons for their complexity include, but are not limited to:

- **Large size and teams**: BHPs can have more than 100 or even 200 million lines of code, even though "70-85% of projects with over 1 M LOC fail". [11- Trendowicz, Jeffrey, 2014, pg.51] "The Standish Group estimates…No single IT project with a budget of above $10 million has ever succeeded on all three [budget, functionality and timeliness] parameters" [12.5- ITworldcanada, March 2001] Their development and management teams are orders of magnitude larger than the size (about seven and not more than 20) of the teams we know how to manage and total more than the number of people (about 150) who can work together productively and be lead and aligned effectively. According to Sauer et al (2007) [11- Trendowicz, Jeffrey, 2014, pg51] "Above the 200 person-year, no successful project was found (in the UK)." What is equally problematic is that team building is not considered to be an imperative for BHPs, because it is not considered to be imperative for other types of projects. So, the many aggregations of people working together are just groups, not teams, and these many large groups of SMEs are managed, not lead, and do not align to one another.

- **Inadequate governance**: Most BHPs do not create a customized project governance structure. Yet one, that is customized to the organization's culture and development environment is needed to allow effective and quick decisions to be made, accountability for these decisions to be assigned and peoples' skills to be exploited. While the project management skills of the oversight and/or steering team members, who make the project related decisions, are often inadequate for the needs of a BHP, their skills to explain underachievement and promise success is exceptional. Because decision makers are not held accountable for the resources spent, they develop acute 'can't-do-nothing' and 'not my fault' syndrome, quote best of class, 'everybody knows' methods that may not be the best for the BHP, and keep growing in number of members and in ineffectiveness.

- **Users' requests focus**: The development team and/or Business Analysts (BA) who define the business' requirements, are focused on the users' requests and corresponding system's functionality or output that will support the functions of (mostly) the current and (some of) the new operation. Many who are new to development, will focus on the details of how the

process works today (the "As-Is'). As few BAs are accountable for capturing the users' requirements, their list of requests will not innovative the business they do not understand, but attempt to automate. In part, this will result in only 20% of the users' requirements being vital for a project to deliver, and only 50 % of the customers' requested output being useful. Their traditional role is not to ensure that the project enables the vision, the intended outcomes, deliver tools for stakeholders to create value (value-based software engineering) or enable a paradigm shift, even if it arguably should be. They will deny, rather than embrace problems when designing new processes and reluctantly integrate new requirements to the already too many, when they are uncovered during the development process.

- **Complex technology**: The technology the system developers embrace and want to deploy is often based on 'best-in-class' not what is best for the organization and is within its capability to maintain it. This can be due to the vast number of software applications that need to be brought on board to satisfy all the sub-groups or the limited available capability of the organization and the focus on the needed functionality not viability of the system (outcome).

- **Deployment:** BHPs are built on paradigm shifts and enable new combinations, more than facilitate existing operational processes. To make the transition, the organization must deploy the business solution, implement the system that supports it and impart the capability to use the new processes to those who need it. This requires making needed changes to the cross-functional business processes that the BHP will enable, engaging the user community and facilitating users to forget the old paradigm. This mandates a Deployment Strategy that not only requires giving users training, but also preparing them with the needed capabilities.

- **Skills of (sub-project) project managers**: Having PMs for each of the several sub-projects, causes several problems and challenges. For one, having many sub-projects increases the problem of communications and miscommunication, for another, it requires more administrative overhead. While the project management competence of these sub-projects' PMs is necessary it is not sufficient, for they not only have to be PMs who manage, but also leaders with the soft skills [2- Janak, 2011] to build their team and be part of the larger BHP team. Unfortunately, many only focus inwardly on their sub-projects and do not report problems that do not affect their sub-project but only impact the BHP. In this way, they provide inadequate interface to other sub-projects. Finally, they work on tasks sometimes in parallel to other tasks in other sub-projects, even when one depends on another, making the portfolio of sub-projects difficult for the PL to align. Finally, many PMs leave their sub-project before the whole project is complete causing additional overhead and making the portfolio of sub-projects next to impossible for the PL to align.

- **Methodology**: BHPs can not be developed using a standard, pre-development defined, 'water fall' systems development lifecycle (SDLC) methodology. Because the details of their business solutions constantly change or evolve based in part on how their system is developed, the final requirements are not known at the start of the design phase. The BHP development methodology has to be a variation of an iterative process, customized for the project, the development environment, the available organizational skills (organizational capability), the organizational capacity and how much innovation the solution is to enable.

BHPs have to be developed while managing requests for changes to the existing requirements (rather than with Requirements Change Management), proactively and continually identifying, deciding on, dealing with and reporting risks (rather than having Risk Management) and dealing with the impact of miscommunications between sub-projects and communicating with executives for quick decisions (not with Communications Management and Reporting Management). While many organizations state that their processes are effective, that does not make them so. There are more ways to justify ineffectiveness [13- Miller, 2013] than there are ways to measure it, and few projects measure their performance. BHPs are challenging projects and only outstanding mostly unmeasured performance can mitigate the impact that they create on the organization, and hence in turn on the project.

- **Escalating cost**: Partly because the design of a BHP's business solution constantly changes, and because its component parts are more intricately interconnected or 'tightly-coupled' [Constantine] than those in large projects, and because many BHPs are intentionally undervalued in what they will cost, the total cost of a BHP will spiral to over 500% and may be to over 5,000% (e.g. Canadian Fire Arms Registry; Canada.ca) of its initially declared development budget. The total development cost will only be about one third of the total cost of the project as over half as much (unbudgeted) will be spent on project administration, requirements management and team management not project development, (about 30% of the total project cost ends up going to aligning and managing the teams according to Ebert and De Man, 2008 [11- Trendowicz, Jeffrey, 2014, pg.57] so cost accounting can significantly misguide executives. In addition, and as expected, the cost of development escalates more when a less realistic, i.e. much smaller than expected, initial budget is allocated (this has two implications that will not be explored here. The first is due to mathematics i.e. dividing by a smaller number yields a larger result. The second is that a larger number results in more effort being spent to stick to the commitment). As the organization most often is incapable of coping with the size of the spiralling increase in cost, especially as at times the earned value is simultaneously decreasing (see Canada.ca in examples), and because they are not able, or in most cases willing to calculate neither the total nor the final cost, much additional effort and cost is spent on debating the future of the BHP.

The situation is not hopeless for the determined. The project management profession has accumulated a great deal of knowledge about managing projects so all we need to do is use it. The following are some of the immutable laws of developing and deploying BHPs, that the astute organization and BHP PL has to be guided by.

Managing a very large project requires skill: "The project leader has been found to be one of the most (if not the most) critical factors in project success." (14- Jiang, Klein, Chen, 2001) The projects of those to whomever this is not obvious, is in jeopardy. The challenge is to evaluate and admit PL capability.

Define explicitly the problem, what constitutes success and the organization's priorities: Elaborate *the solution* the organization expects to deploy to lead it to a *successful outcome*. When success is not measured against this, its achievement is questionable. Act on explicitly defined (verbal definitions are only worth the paper they are written on) *organizational priorities*

that number no more than George Miller's seven (plus or minus 2), and remember that delivering the project on time, on schedule and within budget are not amongst the priorities. Action requires someone to decide and someone to do something, not to discuss and analyse what needs to be done. Investigating, analysing, assessing, reporting and asking for more details are steps in this process but are worthless if they become the result rather than the means to an end.

Reduce the project's size: While the need to reduce the size and hence complexity of the BHP is obvious, how to do it is an art and beyond the present objective. However, it can include delivering only (mostly) *vital functionality*, as most BHPs are designed to deliver twice the functionality that is going to be used. When the unnecessary functionality is removed, assemble the BHP out of the sub-projects that will deliver the needed functionality and focus on the relationships between these sub-projects to align them and produce outputs that together will enable the solution and the desired outcome. Keep the size of the project team, that includes the total number of developers, stakeholders and oversight team members (those who coordinate, advise, oversee but do no develop) to less than 150, and the size of the teams to under G. Miller's seven (arguably 20), as the productivity of a team decreases beyond that size. If the project is seen as too complex, it is, and this cannot be later used as a reason for failure.

Customize the Development Methodology: The development methodology has to be a *customized* version of any standard one or framework adopted by the organization. It has to be based on the available leadership, the project governance that delineates the authority of the PL, and defines roles, responsibilities, accountabilities, and integrity, is based on the existing capacity and capability of the organization as well as on the project. Hold up the vision to illuminate the direction and the 'bridge' or solution to reach the goal. Assign the required resources at the right time and change the schedule when necessary. Finally, develop and implement the project *faster* than the initially accepted schedule, as the longer the project takes the more errors it will have and the more likely it will have major deployment problems. We should remember Peter Senge's well-known quote paraphrased as: "*Every project management methodology and development environment is designed to attain the results it is getting*".

Build SME Teams: Do not take the skill of the selected project team members for granted. Select developers with the skills and skill levels that will be needed because it matters. A BHP must have SMEs who are skilled, accountable, engaged and committed. So, engage them to apply their skill(s) in teams with an effective governance structure.

Get the needed development team members: It is harder to determine prior to project design and initiation, how many people will be needed to develop the project, when will they be needed. It is harder still to get executives to assign and/or procure the needed rather than the available (cheaper) resources. Because at the start, the BHP is not as well defined as a large project and because its scope and requirements evolve as the solution and possibilities become clearer, the number of people and their skill levels will influence the schedule. This is not the case in our standardized methodology that assumes the right people with the needed skill level are engaged and available to the project. Without adequate attention, the BHP ends up with an inadequately large development team, or a much larger team than needed or with team members having

inadequate skills resulting in slower than planned development and a delay in the schedule. If in addition the group that oversees the project but does no development is large and is authorized to manage the project, adherence to the schedule will not be possible.

As some people have more knowledge, education, engagement, love for and experience in some subjects than others, assigning the required resources is not a 'nice to have' or 'luck' but a necessity. It influences the total cost of the project and the schedule. Working with people who are below average performers or who perform above average is not a problem. Not taking their performance level into account when assigning work tasks and scheduling, is. The slow performers will be slower when tasked above their capability, and the 'super star' will be frustrated and/or leave if his/her outstanding performance is impeded or not recognized. While often we are reminded that there is no 'I' in 'team', in the case of BHP sub-project teams, there has to be.

The technology: Due to their size, the technology needed by BHPs is often too complex for the organization to develop and maintain.

Executives' role: Examine and validate how to and select a PL not a PM to take accountability for the project and help define the project governance that will authorize the PL to make project related decisions. BHPs need PLs with exemplary leadership skills, knowledge and experience and personal integrity to build a development environment and a team. PLs need exemplary communications, relationship building and expectation management skills and have to hold up the vision to illuminate the direction. Elaborate on the value executives bring to the project and who has accountability to authorize escalating costs. "poor management can increase software costs more rapidly than any other factor." [12- Barry Boehm, 1981]

Institute a Customer-Centric Requirements Change Management: Change the requirements list, not the requested list, when the customer has a better idea. Assure that the change adds value and leads to better outcome that the requestor is accountable for, as well as the additional cost the change implies and that the executives can afford the implications.

References

1- Joe McKendrick, *The bigger the system, the greater the chance of failure*, ZDNet, Service Oriented, November 3, 2010

2- Pete Janak, former CIO of TRW and Delphi and now principal of PHJ Associates, large [system] implementations projects fail due to "Ignorant and/or incompetent management," Information Week, June 21, 2011.

3- M. Gibbs, *Why your next IT project is doomed*, NETWORKWORLD, May 9, 2013

4- Dr. W. W. Royce, *Managing the Development of Large Software Systems*, Proceedings IEEE WESCON, August 1970."

5-Nadim F. Matta & Ronald N. Ashkenas, *Why Good Projects Fail Anyway*, Harvard Business School Publishing, September 2003 Issue

6- Dr. Paul Dorsey, *Top 10 Reasons Why Systems Projects Fail, Dulcian, Inc.*, 2005

6.2- Capers Jones, *SOFTWARE ENGINEERING: THE STATE OF THE ART IN 2007***,** http://www.spr.com **2007**

6.5- the Thomsett company 2000, *Managing large projects: a special case*, argued that capital investment costs are proportionate to development costs and that the development cost is proportionate to the management and deployment cost.

7- DeAnne, Aguirre, Rutger von Post, Micah Alpern, *Culture's role in enabling organizational change,* Strategy &, 2013

8- THE OTTAWA CITIZEN, March 1, 2012

9- M.Bloch, S.Blumberg & J.Laartz, Delivering large-scale IT projects on time, on budget, and on value, McKinsey, 2012

10- Office of the Auditor General of Canada (OAG) *What prevents large IT projects from being successful*?, 2011

11- Adam Trendowicz, Ross Jeffrey, *Software Project Effort Estimation*, Springer, 2014,

12- Barry Boehm, Software Engineering Economics, Prentice Hall PTR, 1981

12.5- ITworldcanada, March 2001

13- Ken Miller, *We Don't Make Widgets*, Governing Books, 2013

14- J.J. Jiang, G. Klein, H.G. Chen, *The Relative Influence of IS Project Implementation Policies and Project Leadership on Eventual Outcomes*, Project Management Journal, September 2001, Vol 32

About the Author

Charles Villanyi Bokor

Ottawa, Canada

Charles Villanyi Bokor is a Strategic Management Consultant focused on *Leading to Better Decisions*. Principal activities include Business Transformation, Problem Project Recovery & Leadership, Strategic Planning. Charles works mostly in Ottawa but has successfully completed assignments in Florida, Wales, Malaysia, Sweden and Australia, and was key-note speaker in Johannesburg South Africa and Victoria BC. Formal education includes an Executive Development and Diploma in Management (McGill University), M.Sc. Mathematics (Université de Grenoble, and U. de Montréal) and B. Sc. Mathematics (Concordia University). He was: Program Director of the Corporate Performance Management Program, Sprott, Carleton; Director of IS/IM at Royal Trust; and at Northern Telecom; CMC; CMC Board Member; PMI-OVOC Board Member; Governor of ICCC; is ITIL Certified and a TBS Independent Project Reviewer. Charles can be contacted at villanyibokor@gmail.com