Measuring Information Technology (IT) Project Performances in Texas: House Bill (HB) 3275 Implications *(a position paper)*

Abstract – Texas' usage of IT is big and getting bigger, but past project performances have a "checkered" history. In June 2017 HB 3275 became law in Texas. It requires state agencies to improve the measuring and monitoring of large IT projects to collect and report on performance indicators for schedule, cost, scope, and quality. If these indicators go out of bounds, more intense scrutiny is then triggered, potentially requiring corrective action. These indicators will be made visible to the public via an online, user-friendly dashboard, and will be summarized annually in a report to state leaders. This new law facilitates the early detection of troubled projects, and helps establish baselines for improvement initiatives. This position paper discusses the implications and challenges of implementing this new law for state and agency IT leadership.

Context

Our daily world relies heavily on measurements, and without them, there would be chaos. The same is true in the world of IT systems and projects.

The following quotes set the stage for meaningful measurement of IT projects:

- *"When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind . . ."* - Lord Kelvin, 1891
- *"If you cannot measure it, you cannot control it. If you cannot control it, you cannot improve it."* – Peter Drucker
- *"In God we trust, all others must bring data."* - W. Edwards Deming
- *"It is impossible to develop measures of progress if we do not know where we are trying to go"*
- *"With proper IT measurement discipline, we can expect: improved visibility into project performances, the prevention of future disaster projects, and the stimulation of continuous improvement initiatives in all state agencies"*

Measurement is incredibly important to state agency IT projects. But like many important things, it is a subject that is not fun to talk about. Measurement is a boring subject. Collecting data and looking at charts and graphs of that data are not exciting for most of us. Rather than collecting statistics on everything, it is important for an organization to have a small yet balanced set of vital metrics used to drive decision-making about progress, performance and improvement.

Existing industry standards[3,9] identify the base set of measures that all IT projects should report and track. There is widespread agreement that a minimal set of project level measures should be about: *cost, schedule, scope* and *quality*. Additional measurements may also be desirable (e.g. risk).

In Texas state government agencies, it has been anecdotally observed that:

- there currently exists an inconsistent and incomplete understanding of the size, scope and

true status of Texas' largest IT projects,
- state spending on major IT projects[1](aka major IR projects or MIRP) lacks appropriate oversight, and
- better tracking is needed to provide Legislators, the public and other stakeholders accurate insight into the state of the existing IT project base.

On June 18, 2017 HB 3275[2] became law in Texas, taking effect on **January 1, 2018**.

What the HB 3275 law requires

The Texas Government Code, Section 2054.151, states that "the legislature intends that state agency information resources (IR) and information resources technology projects will be successfully completed on time and within budget and that the projects will function and provide benefits in the manner the agency projected in its plans submitted to the department (DIR) and in its appropriations requests submitted to the legislature."

HB 3275 requires the enhanced measuring and monitoring of major IT projects[1] by the state's Department of Information Resources (DIR)[5] and the state's Quality Assurance Team (QAT)[6]. QAT is primarily responsible for the monitoring and reporting, with the assistance of DIR in defining the reported performance indicators and implementing an online dashboard. In the Texas statutes IT projects are often referred to as IR projects. In the remainder of this paper we will use the more industry standard term, IT.

QAT is now required to:
- Monitor and report on performance indicators for each major IT project for its entire life cycle. The enhanced performance indicators must include measurements of: *schedule, cost, scope, and quality*.
- Place a project that they determine is not likely to achieve their performance objectives on a list for more intensive monitoring.
- Closely monitor monthly reports for those projects identified as needing more intense monitoring based on the criteria developed by DIR, to determine whether to recommend to DIR's executive director the need to initiate corrective action for that project.
- Enhance their annual report to include the current status of each major IT project[1] and its set of performance indicators.

DIR is now required to:
- Develop the performance indicators QAT is required to monitor using applicable IT industry standards in adopting those rules; specifically including at least indicators of: schedule, cost, scope, and quality, for each project's entire life cycle.
- Create and maintain on the DIR website a user-friendly data visualization tool that provides an analysis and visual representation of the performance indicators for each project.

Agency management will now need to step up and report on the basic four indicator areas for all major IT projects. Having such data available will enable the state to better monitor and achieve project performance objectives, as well as achieving IT organizational improvement targets.

HB 3275 is part of a broader strategy to improve the acquisition of IT capabilities across all state agencies; with a strategic goal for each to become a highly efficient, effective, transparent, accountable and a consistent provider of excellent customer service. Innovative and aggressive use of IT is posed as a primary way to achieve that goal.

The Implications for IT Project Measurement in all state agencies

There are implications of the new law that will affect all state agencies engaged in performing major IT projects. They will need to understand and then implement the measurement mechanisms to collect the base data that will support the reporting of the enhanced performance indicators. The challenges will include creating the definitions and new measurement mechanisms for the data to be collected and reported; especially for those factors (scope, quality) which have not been reported on in the past.

What we know now about large IT projects in Texas Agencies

QAT currently summarizes and reports annually on agency reported measures of major IT projects budget and duration[4]. Several interesting excerpts from their December 2016 report are:

1. From December 2015 to November 2016, the state's monitored portfolio included 76 major IT projects totaling $1.4 billion in estimated costs.
2. Only 47 of the 76 projects were rated, as the reporting threshold required a project to be 30 percent or more complete as of November 2016.
3. The rating of each project is determined by comparing the current cost and duration estimates of a project to its initial cost and duration estimates. Those are then graded at three levels: 1) within 10% of original estimates of cost and schedule (green; 16), 2) exceeding estimates of cost OR schedule by more than 10% (yellow; 19), and 3) exceeding estimates of cost AND schedule by more than 10% (red; 12).
4. Eight projects were canceled since the previous annual report.
5. The rate of successful project performances declined in 2016 relative to 2015.
6. It was generally observed that projects with a shorter development schedule (less than 28 months) are meeting their initial cost and duration estimates at a higher rate relative to projects with longer durations. Longer projects with initial estimated costs of more than $10.0 million were less likely to be implemented within budget and duration targets.

HB 3275 puts the basic measurement requirements in place to determine whether or not a major IT project might be in trouble, and thus in need of a direction adjustment. HB 3275 increased the emphasis on IT performance measurement, which is a way to make sure that what matters is measured, that what is measured really matters, and that results reported are credible.

Warning signs that an IT Project Might be in Trouble

There are a number of measurable signs that an IT project might be in trouble. These are:

1. **Ratios Trouble**: Cost ratios and schedule ratios are metrics that allow organization leaders to measure budgeted time and money verses money and time actually spent. Without these

metrics, all you have to rely on is the accuracy of the communication you receive from project teams about performance.

2. **Milestones Aren't Met**: *Small*, *discrete*, and *often*, are the guidelines for setting milestones of a successful project. Although this may seem obvious, it's surprising how many times lack of achievement is ignored or rationalized.
3. **Scope Changes**: A common approach to shoring up a lagging project is to change the scope of the work product. Eliminating features or relaxing requirements is not uncommon, but when project teams are doing this *because the project is in trouble*, it's a huge warning sign of danger ahead. The more frequent and later in the timeline, the bigger the concern.
4. **Poor quality** deliverables: When poor quality code is delivered, that is a sure sign of the lack of engineering discipline applied to development, or a lack of quality standards being applied. For non-code deliverables (requirements, architecture, designs, test plans, technology roadmap, etc.) rigorous review mechanisms (e.g. Quality Assurance, or "QA") must be in place to measure against defined acceptance criteria – Quality just doesn't happen by itself, it must be *planned* into the work processes used.
5. **Unmanaged risk**: Most large IT projects are highly risky and thus require a suitable risk management process. For example, a project with a "failure is not an option" mantra ignores that a system might fail to meet its objectives. Not having a risk management plan or process is a sign of trouble, as is, unmeasured risk potential.
6. **Lots of Overtime**: A project running on schedule should have little or no overtime. Overtime is often a quick fix, but it leads to more mistakes. It also leads to poor employee health resulting from too much caffeine, too many late nights, and too much junk food. "Death march" projects[12] are the worst offenders.
7. **Diversion of Resources**: When people are pulled from one project to work on another it could be a sign of trouble. If you've budgeted your staffing resources properly, a few hours here and there on a troubled project can quickly add up and cascade down, endangering other healthy projects. "The good rats jumping off a sinking ship" is a big red flag.

The earliest signs that an IT project is in trouble are harder to measure objectively, but are relatively easy to spot. They include:

1. **Lack of Interest**: Whether it's a lack of interest within the project team or among stakeholders, a lack of interest is often demonstrated by people not showing up for meetings, a lack of active participation and feedback, or a poorly energized user base. Especially telling is the lack of interest by executive management stakeholders.
2. **Poor Communication**: If few stakeholders are communicating, including management, team members, and end users, there could be a problem. Lots of meetings that accomplish little, and complaints about issues "going into a black hole", are common problems on projects without adequate issue and action tracking.
3. **Lack of Velocity**: Projects should always be moving forward. The best way to keep a good velocity is to divide your project into small deliverables at frequent intervals (milestones). Agile projects are especially good at this.
4. **A "No-Bad-News" Environment**: Nobody likes to be the bearer of bad news, but sometimes organizations need to face reality. This includes project team members who don't want to be the messenger and organization leaders who tend to shoot the messenger. If there is not an environment where the communication is honest about "reality," projects tend to fail more frequently.
5. **Lack of defined and measurable objectives**. A *performance measure* is an indicator of progress toward achieving a goal; it is a "you are here" on the map of progress. A common

problem cited in research—and often seen in practice—is that a project's goals/objectives have not been identified. *It is impossible to develop measures of progress if we do not know where we are trying to go.*

6. **Unprioritized objectives and requirements**. Without prioritization into must-haves and need-now's, or at least an expectation matrix of how scope, schedule, cost, and quality trade-offs should be handled, projects can spend too much time and energy on less significant work.
7. **Lack of a *baselined* workplan**. The purpose of planning is to model how the work should go to reach the project's goals/objectives. But if not baselined early, the model's predictive value is severely limited, and incremental slippage begins to occur.
8. **Excessive project management overhead**. Any project with oversight incurs overhead, but large projects can be overburdened with "too many cooks in the kitchen", as evidenced by multiple managers in the same role or organization chart box, or excessive and/or unproductive meetings. The more humans involved, the more complex the communications amongst them gets. Lean and flat management structures are best with highly educated and motivated staff.
9. **Poor-fitting development methodology and toolset**. Applying the wrong methodology model to a project is an early indicator of trouble ahead. "Shadow systems", Skunk-Works, and clandestine experiments with technology may be an indicator of ill-fitting tools or methodology, likely to cause trouble later on.
10. **Low process maturity**. No matter what development model is used (e.g. adaptive like Scrum, or predictive like waterfall), higher process maturity organizations have a higher likelihood of project success. Observed rigor with the chosen model can help predict a project staff's process maturity early in the project timeline.
11. **Overly complex/deficient system architecture**. Every project needs an architect - someone with the overall vision for the structure and function of the final work products. This vision needs to be articulated at a level of detail sufficient for the project staff to see how it all fits together. Projects with overly complex and/or deficient systems architectures typically lead to poorer results.

Measurement Implications and Objectives

The remainder of this paper uses more-formal, standard defintions[8] for the terms related to IT project performance measurement. These are:

- A ***metric*** is a measure of some property of a piece of an IT system or its specifications. The term "metric" is often misused interchangeably with measure and measurement.
- ***Measurement*** is the process by which numbers or symbols are assigned to attributes of entities in the IT world in such a way as to characterize them according to clearly defined rules. More formally, measurement is a mapping from the IT world to the formal, relational world (mathematical)
- A ***measure*** is the number or symbol assigned to an entity by this mapping above in order to characterize an attribute. It is a quantitative indication of amount, dimension, capacity, or size of product and/or process attributes.
- A ***performance indicator*** is a judgment based on a measure or combination of measures (e.g. the size of a module as *too big*). Once base measures have been developed, performance indicators derived from them can be obtained.

Measures and indicators can then be used as descriptors, predictors or prescriptions.

Having quantifiable measurements allows us to:

- Reflect and measure achievements on critical success factors of a project
- Base them on beforehand agreed measurement definitions
- Act as an indicator of progress towards the goals of what is being measured
- Reveal a high-level snapshot of the what is being measured

The attributes of effective IT measures are:

- Simple and computable
- Consistent and objective
- Consistent in their use of units and dimensions
- Programming technology (e.g. language) independent
- An effective mechanism for performance feedback

HB 3275 requires performance indicators for cost, schedule, scope and quality. Additional indicators are also reasonable to consider for more specific situations (e.g. risk or agile). Having just enough measurements is important to the effective and efficient delivery of IT systems.

Measuring and Monitoring of Major IT projects for Quality

Typical rework rates exceed 60% of an IT developer's time; so quality is usually an issue to be managed. IT developers informally consider quality to mean that the system meets specifications, runs efficiently, doesn't crash, follows standards, uses current technology and techniques, is secure and can be modified easily. IT quality is an engineering concept that can be defined objectively and thus measured by relevant factors.

The measurement of IT Quality, whether focusing on the Customer, the Product and/or the Process, requires the collection and analysis of certain data, stated in terms of defined measures. Quality measures are usually reported with respect to articulated quality goals, and represent both internal and external characteristics of the emerging system. Standard production oriented measures are: Mean Time To Failure (MTTF), Mean Time To Repair (MTTR), availability, security violation rate, and source code complexity metrics.

There are IT industry standards that can be used (e.g. ISO/IEC 25000:2014, CISQ[13]) to define an IT quality model for each project. And there are measurement tools for the common model components (e.g. problems, defects, vulnerabilities, architectural flaws, technical debt, cost of rework, etc.).

Since this is a new area of required measurement, particular attention will be needed to define the framework for effective measurement and derivation of reported quality performance indicators.

Measuring and monitoring IT project scope

Establishing a project's scope is that part of planning that involves determining and documenting a list of specific project goals, deliverables, tasks, efforts and deadlines. These define the boundaries of the project, establish responsibilities for the team, and set up processes for how completed work will be verified and approved.

In general, scope should be crisp and well defined, and derived from the business objectives. Identifying features and functions that are *specifically excluded* from work products and deliverables can be extremely useful, as it helps sets firm, clear boundaries, helping to curb scope creep and gold plating. The initial version is often referred to as the Scope of Work , suitable for inclusion in a Request For Proposal for external contracting.

The scope definition also provides the project team with guidelines for making decisions about change requests during the project (e.g. Expectations Matrix). It is natural for parts of a large IT project to change along the way (after all, projects are *all about* change – from the way things were in the past, to the way they will be in the future). So, the better the project has been "scoped" at the beginning, the better will be the baseline to manage change against. How changing scope is managed is a prime determinant of success. This is especially true as organizations move towards more adaptive agile and lean IT projects.

Since scope is a new area of required measurement, particular attention will be needed to define the framework for effective measurement and derivation of reported scope performance indicators.

There are several industry standards to consider using in this area, depending upon the type of development lifecycle chosen (e.g. function points, enhancement points, feature points, user story points, requirement statements, epic/thematic specifications, etc.).

Measuring and monitoring project cost

A project's cost is a function of resources consumed over time. The project work plan to achieve the desired scope maps this consumption against the derived schedule. The work plan will drive the cost model for the project, which becomes the **Performance Measurement Baseline (PMB)**, representing the approved costs for the project over time.

At any point in any project's timeline, project sponsors would like to know:

What did we <u>get</u> for what we <u>spent</u>?

To monitor cost consumption when the project is underway, three key values are needed for each measurement point:

- **Planned Value (PV)**: the authorized budget applied to <u>scheduled work</u> (aka "the work plan").
- **Earned Value (EV)**: the measure of <u>work performed</u>, expressed in terms of the budget authorized for that work.
- **Actual Cost (AC)**: the realized <u>cost incurred</u> for the work performed on an activity during a specified time period.

The total expected cost for the project is the **Budget at Completion (BAC)**. The total budget includes risk contingency reserves (for the identifiable risks) and management reserves (for unforeseen work within the scope of the project).

When the project is underway, actual staff effort must be captured in some way (per unit of time) to be recorded as Actual Cost (AC). At any point in time, this can be compared to the Planned Value (PV) expected for that same time period, to measure Cost Variance (CV). Recognition of cost variances early in the project allows for early (and smaller impact) adjustments to the project work plan.

To monitor a project *effectively* at <u>any given point</u> on the project timeline, answers to the following questions are desired:

- What work is scheduled to have been completed?
- What was the cost estimate for the work scheduled?
- What work has been accomplished?
- What was the cost estimate for the completed work?
- What have our costs been to date?
- What are the variances?
- What is the total cost/effort growth?

Commonly used cost measures for major IT projects are shown in the table below.

Metric	Name	Definition
PV	Planned Value	Planned cost of work to be done (at a point in time). <u>Answers the question</u>: How much work *should be done*?
EV	Earned Value	Value of work *actually* accomplished (at a point in time). <u>Answers the question</u>: How much work *is done*?
AC	Actual Cost	Cost of work completed (at a point in time). <u>Answers the question</u>: How much *did* the "is done" work cost?
BAC	Budget at Completion	Total planned value of the entire project (baseline). <u>Answers the question</u>: What was the *total job* supposed to cost?
ECAC	Estimated Cost At Completion	<u>Forecast</u> of what the project will *actually cost* (at a point in time). [ECAC = BAC / CPI] <u>Answers the question</u>: At this point in time, what do we *now* expect the *total job* to cost at completion ?
CV	Cost Variance at a point in time	A calculation of Earned Value less Actual Cost (at a point in time). [CV = EV – AC] <u>Answers the question</u>: How much over/under budget are we (at a point in time)? And how fast is it changing?

VAC	Variance at Completion	A forecast of Budget At Completion less Estimated Cost At Completion (at the project end). [VAC = BAC – ECAC] Answers the question: How much over/under budget do we expect to be at the project end?
CPI	Cost Performance Index at a point in time	An indicator of the cost efficiency of budgeted resources expressed as the ratio of Earned Value to Actual Cost (at a point in time). [CPI = EV/AC] (≤1 is good) Answers the question: How closely are we following our cost consumption plan (at a point in time)?
TCPI	To-Complete Performance Index	The future CPI that you must achieve for the remaining work if you want to complete it within the given budget (at the project end). [TCPI (if under budget) = (BAC – EV) / (BAC – AC)] [TCPI (if over budget) = (BAC – EV) / (ECAC – AC)] Answers the question: What rate of performance must we achieve to finish on budget?

Measuring and monitoring project schedule

Schedule performance is a measure that is reported on a regular basis to the project sponsor and key stakeholders. It is normally incorporated into project status reports. In most cases it is a discussion topic during Project Steering Committee updates. For each of the summary tasks (project phases) or key milestones reported, the following information is provided on a regular basis:

- **Planned (Baseline) vs. Projected (Current) Finish Date** – Both of these data points are obtained directly from the schedule for the summary task or milestone.
- **Variance** – It is helpful to report the variance in terms of *days*, and as a *percentage of the total duration* for the summary task or milestone. The variance is also obtained directly from the project schedule, and the variance percent is calculated based by dividing the variance by the total duration.
- **Trend** – Comparison of the variance to previous reporting periods. Is the metric trending positively, negatively, or holding steady?
- **Comments** – Explanation of the variance (root causes), and the next steps that are planned to monitor / manage the variance (corrective action) including project impacts that have been initiated).

The information reported might be a summarization of several explanations documented during the schedule analysis process. If a variance were formalized as a project impact, it would by default be reported to the project sponsors through the normal management control process.

The accuracy and credibility of the schedule measures are highly dependent on the following critical success factors:

1. The original baseline created, approved and saved in the project management tool, must be solid. If the baseline is not strong, comparisons to actual results will be difficult to explain and manage throughout the entire project life cycle.
2. The schedule performance analysis is only as good as the schedule you are analyzing – the current schedule must reflect the current "reality" of the project. Therefore, the schedule must be maintained and managed in a manner that keeps it in synch with the way that work is being performed by the team. Due to the uncertainty of the future, it is often wise to plan to re-plan during the course of a project, based on events.
3. As project impacts are approved and implemented, the project schedule (or the portions impacted) must be re-baselined. If the schedule is not re-baselined, the original variance will still be included with any new variances that occur (confusing the on-going schedule analysis and reporting process). However, an organization should have pre-defined criteria for resetting a baselined schedule. If not, frequent re-baselining can occur, masking real slippage. This can show up readily in a slip chart of successive project baselines.

Schedule metrics are directly related to the cost metrics above and are part of the Performance Measurement Baseline (PMB). The most common schedule metrics are:

Metric	Name	Definition
SV	Schedule Variance at a point in time	A calculation of Earned Value less Planned Value (at a point in time). [SV = EV – PV] (≥0 is good) Answers the question: How early/late are we (at a point in time)? (expressed as the sum of *work value* we should have earned at this point in time)
SPI	Schedule Performance Index at a point in time	An indicator of the schedule efficiency of budgeted resources expressed as the ratio of Earned Value to Planned Value (at a point in time). [CPI = EV/PV] (≤1 is good) Answers the question: How closely are we following our schedule consumption plan (at a point in time)?
ETAC	Estimated Time At Completion	Forecast of when the project will *actually end* (from a point in time). [ETAC = Original-End-Date / SPI] Answers the question: At this point time, when do we *now* expect the *total job* to be completed?

Measuring and monitoring other project aspects beyond the minimum requirements

A baselined work plan with costs and schedule plotted over time represents the project manager's best model of how the work should proceed. But the future is uncertain, so risks to that work plan must be brainstormed as much as practical. This starts with examining *all assumptions* made in project planning – what to do should an assumption not be met. The actions to be taken constitute **contingency reserve**.

But it is impossible to anticipate everything, so a **management reserve** must also be created, to accommodate the unforeseen events. These reserves apply to both money *and* time, as they represent buffers against the unknown.

During development of the work plan, it is recommended when estimating effort (regardless of the method(s) used) to *also* capture a **confidence factor** that represents how solid the estimated time/costs are. This information is crucial to sizing the amount of reserves (buffers) that are needed to accommodate the associated risk with the estimated work.

It is also recommended that the calculated risk reserves be factored into the work plan such that its consumption can be readily tracked. So, during project execution, reported Actual Costs (consumed effort) and Estimated Work Remaining for any given task (what's left to do), could allow for direct reserve adjustments in the work plan. This puts reserve management under control of the project manager, rather than it being hidden in the work plan under the task-owner's control.

Some tasks in the work plan will likely take longer than expected, and some will be shorter. This is normal as a work plan is really just a collection of *estimates*. But these under/overs can be added to / subtracted from the reserves for a particular deliverable as the project progresses. Then a good predictor of task (and project) completion can be viewed by charting reserve buffer consumption against planned work products (PV). Often, a "fever chart" (percentage plot of earned milestones vs. reserves consumed to complete them) with red-yellow-green zones indicating the level of attention and action needed serves as a good tracking instrument.

Reserve management tracking, in conjunction with the other cost and schedule metrics discussed earlier, provide a quantitative way to track and report on project performance, informing *anticipatory* rather than *reactionary* action by management.

On IT projects, measuring risk boils down to determining *what will it cost if an identified adverse event happens?* For that reason, we try to express all risk in terms of cost. Everything ultimately costs money. Time costs money, so schedule slips can be converted into dollar equivalents. Once everything is in the same units, a project's risk exposure can be ranked objectively.

The expected loss is the total likelihood of the adverse event occurring (Event Probability times Impact Probability) times the total loss. Risk management[10,3] is a well-understood and mature process that can be applied to large IT projects. It is also possible that risk measurement could be incorporated into the basic four indicator areas.

For agile and lean projects, the additional measures are usually value added, lead-time, cycle time, team velocity, open/close rates and waste.

- **Value added** - The relationship between the software, the desired outcome, and this metrics forms the value hypothesis for the feature (or system, or story, or upgrade, etc.). Business value points are assigned to system features and attributes.
- **Lead-time—**how long it takes to go from concept to delivered software. Lead-time includes cycle time. **Cycle time** is how long it takes to make a change to the software system and deliver that change into production. Teams using continuous delivery can have cycle times measured in minutes or hours instead of months.
- **Team velocity—**how many "units" of software the team typically completes in an iteration (a.k.a. "sprint"). This number is used to plan iterations and track productivity.
- **Open/close rates—**how many production issues are reported and closed within a specific time period. The general trend matters more than the specific numbers.
- **Wasted time and effort** – Time/effort that is non-value added in a lean context. This is ultimately caused by uneven workflow and unreasonable demands. This includes such aspects as: overproduction, excess work in progress, transportation costs, excess motion, delays in process and, most importantly for IT, **rework**.

Next steps

It will be important for agency CIO's and IT Leaders (e.g. IRMs, IT PMOs, IT project leaders, vendor contract leaders) to begin the process of defining more precisely what these performance indicators mean; and laying out the framework for collecting and reporting on IT performance measurements. DIR will likely lead that initiative, but it would be best if all levels of state agency IT management would participate in the definition and development of these needed measurements. One option would be to hold a topical forum (e.g. Texas IT Forum) on this subject with all stakeholders present to identify/define the base measurements and performance indicators needed for cost, schedule, scope and quality.

Goal-oriented measurement is the definition of a measurement plan based on explicit and precisely defined goals that state how measurements will be used. In addition, explicit models have to be defined to support the derivation of measures from the goals in a traceable and unambiguous manner. Goals-Questions-Metrics (GQM) methodology[7,8] has been found quite useful for that purpose.

GQM is a known technique for identifying suitable measurements to collect during an IT development project. It is based on the observation that data collection for its own sake is usually a waste of time and effort. Hence, it is only worthwhile measuring things to satisfy specific goals. The technique starts by identifying some high-level goals related to the project (e.g. "keep the project on schedule", "get all the severe bugs out", etc.). For each goal, it identifies a number of questions that need to be asked if the goal is to be achieved (e.g. "how far behind schedule are we?", "how many severe bugs are left?"). Then, for each question, it identifies measurements that can be taken to answer the question. A question may be related to more than one goal, and a measurement may help to answer more than one question, so the result is a hierarchical model. See[7,8] for more information on how to use this technique.

The above analysis will then allow us to define a standard dashboard for all large IT projects, which will in turn:

- Facilitate control over measures by providing them all in one place
- Provide a convenient mechanism where the project manager can see and control all the key measures related to his/her project
- Allow for key indicators to be derived from several key elements (areas) concurrently
- Offer a level of transparency for people who are not interested in the detailed numbers

We, as the IT professional community, have the opportunity with this new law to all get on the same page with respect to IT project performance measurement. Let us take that step together.

References

1. A major IT project is typically > $1M and/or longer than a year to complete. See Texas Government Code, Chapter 2054.003 for details. http://www.statutes.legis.state.tx.us/SOTWDocs/GV/htm/GV.2054.htm
2. HB 3275 championed by Herb Krasner, sponsored by Rep. Giovanni Capriglione, https://legiscan.com/TX/text/HB3275/2017
3. *A Guide to the Project Management Body of Knowledge*, 6th Edition; and the *Software Extension to the PMBOK® Guide*. http://www.pmi.org/pmbok-guide-standards
4. Annual Report - OVERVIEW OF MAJOR INFORMATION RESOURCES PROJECTS REPORTED TO THE QUALITY ASSURANCE TEAM, December 1, 2016, http://qat.state.tx.us/pubs.htm
5. DIR website, http://dir.texas.gov/
6. QAT website, http://qat.state.tx.us/
7. *The Goal/Question/Metric Methodology: a practical guide for quality improvement in software development*, Solingen and Berghout, 1999, McGraw-Hill Publisher
8. EE 382v, Graduate Course in Software Measurement, class notes, H. Krasner, 2014, University of Texas at Austin
9. ISO/IEC 25000:2014, https://www.iso.org/standard/64764.html
10. https://en.wikipedia.org/wiki/Risk_management
11. *Earned Value Project Management*, Q.W. Fleming and J.M. Koppelman, PMI, 2006
12. *Death March, Edward Yourdon, Prentice Hall, 2003*
13. http://it-cisq.org/standards/automated-quality-characteristic-measures/