The newsletter of the Massachusetts Institute of Technology
System Design and Management Program

sdmpulse

in this issue

MITsdm system design and management

SDM Best Thesis analyzes costs of software system complexity

By Andrei Akaikine, SDM ’09

Editor’s note: *Andrei Akaikine was awarded the SDM Best Thesis Prize in October 2010 for his thesis, “The Impact of Software Architecture on Product Maintenance Efforts and Measurement of Economic Benefits of Product Redesign.”*

Andrei Akaikine
SDM ’09

“If it ain’t broke, don’t fix it” is the basic principle used to manage complex socio-political and engineering systems. For many engineers, the risk of upsetting a system’s stability outweighs the benefits of attempting to improve it when there is no evident flaw.

However, maintaining a complex system at a “good enough” level of functionality comes at a cost. In systems where stability is paramount, personnel dealing with small emergencies traditionally rely on “Band-Aid” fixes that increase complexity and heighten the risk of destabilizing the whole system with each new modification, however minor.

In most cases, a product’s lifetime maintenance cost overshadows the cost of initial product development by a large margin. Based on the empirical data for the computer software industry, the cost of a fix grows exponentially between each phase of the product’s life cycle. Some sources estimate that activities during the maintenance phase may consume as much as 75 percent to 90 percent of total product lifetime expenses.

Measuring costs associated with maintaining complex computer systems is one of my long-time interests. Having spent more than five years working in product support and sustained engineering at Microsoft, I have seen the effects of product architecture complexity on maintenance costs and overall software system stability. Combining my interests with the knowledge that I acquired in MIT’s System Design and Management Program (SDM), I focused my SDM thesis research on the following three questions:

- Is there a relationship between engineering efforts required for a basic maintenance task and the overall software system complexity?
- Are any components more susceptible than others to change during system maintenance tasks?
- Can the redesign of a system be economically justified?

To answer these and other questions regarding the effects of system complexity on the cost of software maintenance, I designed an experiment

> continued on page 8

sdmpulse

Vol. 6 No. 1 | spring 2011

Publisher: Joan Rubin, SDM Industry Codirector

Editor: Lois Slavin, MIT SDM Communications Director

Managing Editor: Kathryn O'Neill

Contributors: Azamat Abdymomunov, Andrei Akaikine, Augustin Friedel, Sorin Grama, Donny Holaschutz, Ken Huang, Ben Jiang, Alan MacCormack, Rafael Marañón-Abreu, Christine Miyachi, Shalom S. Saar, Yoav Shapira, David Sharman, Helen M. Trimble

Photography and Illustration: Andrei Akaikine, Anthony Do, Augustin Friedel, Eugene Gorelik, Hubspot, Alan MacCormack, Christine Miyachi, Ethan Pease, David Sharman

Design: Stoltze Design Inc.

Layout: Janice Hall, TTF Design

Printer: Arlington Lithograph

MIT's SDM program is cosponsored by MIT Sloan School of Management and MIT's School of Engineering. SDM resides within the MIT Engineering Systems Division.

For further information on MIT's System Design and Management Program, visit sdm.mit.edu.

MITsdm system design and management

Welcome

Welcome to the spring edition of the *SDM Pulse* newsletter. 2011 marks a milestone for SDM—the 15th cohort entered the program this January, bringing with it broad diversity in backgrounds, culture, and experience.

Thinking back to 1996 when SDM started, we lived in a world where iPods didn't exist (they were introduced in 2001), hybrid cars were still a couple of years away from mass market availability, the first version of Java was released, and eBay was brand-new, hosting just 250,000 auctions that first year (more than 2 million visitors now browse the site daily). In the 15 years since, we have not only grown accustomed to the fast pace of change in technology; we expect that it will continue to accelerate. The rapidly changing technical world has created a need to develop leaders that have a strong foundation in systems thinking to manage increasingly complex challenges.

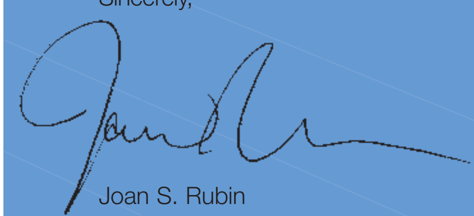
As we think about life without text messaging and iPods, this issue of the *Pulse* has several articles that touch on the complexity of software development in products such as these and the need for a thoughtful approach to rapid iterations of software to manage escalating costs associated with band-aiding code as products mature. Also included in this issue is a series of profiles of alumni who have taken their careers in an entrepreneurial direction. Their success has resulted in technology that is used by our industry partners and is a reflection of the innovation that is being driven not only in these small startup companies, but has been integrated by our alumni in a range of more traditional companies and industries.

I encourage you to take time to review the overview of the 2011 cohort that joined SDM this January. It is both the largest and most diverse group yet, with a tremendous breadth of experience. Unlike traditional engineering or MBA programs, the SDM program has attracted students that have an average of nine years of work experience and a focus to their studies that comes both from career experience and from an understanding of the applicability of the coursework to the companies that they will work for in the future.

As a final note, I am thrilled to join the SDM program as industry codirector. I have spent the last 17 years in the medical device area in roles ranging from marketing to business development. The common thread in my career has been building strong relationships with clinicians, small startups, and large global corporations to improve access to technology and increase value for all. Looking forward to the next 15 years with SDM, I am hoping to expand the connections with industry to build leaders and increase the benefits for everyone who is touched by the program.

I look forward to talking with many of you in the months ahead and welcome your feedback and comments at any time.

Sincerely,



Joan S. Rubin
Industry Codirector
MIT System Design and Management Program
jsrubin@mit.edu

SDM course helps students build a leadership roadmap

By Shalom S. Saar, senior lecturer, MIT Engineering Systems Division

Shalom S. Saar

Editor's note: *In this article, Senior Lecturer Shalom S. Saar describes Leadership: The Missing Link, a required course in MIT's System Design and Management Program (SDM).*

SDM's leadership course is designed to prepare students to become better decision-makers. Students emerge with a richer understanding of their strengths and weaknesses—as well as a roadmap for leveraging their abilities to become more successful leaders.

Too often, students who return to school for upper-level engineering degrees overemphasize their need for technical skills and underemphasize the importance of gaining people skills. Yet the reality is that conflict is a growth industry. As business becomes more global, conflicts proliferate—it's simply too easy to read an unfamiliar environment the wrong way, and different leadership styles prove more successful in different situations. While the technical side of being at MIT is important, the people side is just as important, if not more critical.

The purpose of Leadership: The Missing Link is to enhance each student's ability to lead and mobilize others. Helping students become aware of themselves and their impact on others can increase their level of competency to work through others. To paraphrase the ancient Chinese general Sun Tzu, author of *The Art of War*, if the executive doesn't know himself and doesn't know his opponents, his chances of winning are very low.

> Shalom Saar's course was the ideal capstone for the management aspect of the SDM program. [His] teaching is without parallel. He brings situations to life ... and makes the time spent in this class among the most valuable time of your MIT student-career.
>
> *—Blade Kotelly, SDM '10*

> I am not sure what I'll do after SDM, [but] wherever I go I'm bound to face leadership and communication challenges for which this course was very useful.
>
> *—Avi Latner, SDM '10*

> Shalom Saar is a wise instructor who creates a trustful and interactive learning environment, which together with cutting-edge diagnostic tools helped me redefine questions on leadership, trust, collaboration, authority, conflict resolution, and human interaction.
>
> *—Azamat Abdymomunov, SDM '10*

The course utilizes various instruments, real-world case studies, and simulations. While the first part of the course focuses on understanding oneself, the second part assists students in developing the skills needed to motivate and influence others. We focus on such topics as strategic thinking, leadership styles, personality types, approach to conflict, and emotional intelligence. In addition, we use a 360-degree feedback diagnostic tool that each SDM student is required to complete during orientation.

Students conduct the 360-degree assessment by requesting feedback on 25 leadership competencies from a wide array of colleagues—including supervisors, subordinates, peers. The report they get from this instrument is usually eye opening; it reveals that the perception someone has of himself is not always the perception others have. One woman might think she's a good listener, for example, only to discover that her friends and colleagues think she ignores their views.

By the end of the course, students are able to assess their strengths and weaknesses, but the class is also prescriptive. Students learn how to probe, listen, influence, negotiate, and motivate.

In one exercise, for example, students learn the value of

> continued on page 9

Faculty member outlines value of software system architecture

By Alan MacCormack, visiting associate professor, MIT Sloan School of Management

Alan MacCormack

Editor's note: *Alan MacCormack is a member of the Technological Innovation and Entrepreneurship Group at the MIT Sloan School of Management. He teaches Sloan's core class in innovation and has served as thesis advisor for several students in MIT's System Design and Management Program (SDM), including Andrei Akaikine, SDM '09* *(see story, page 1).*

Over the last 10 to 15 years, even the most traditional of industries have come to rely on software for everything from inventory control to vehicle navigation. The average automobile today has more software than the first Apollo moon rocket. Your garden variety microwave may even have an algorithm for cooking popcorn to fit your specific tastes. Despite this dramatic increase in the pervasiveness and importance of software, however, many companies lack a fundamental understanding of the architecture underlying their code. This problem costs firms millions of dollars per year.

Ask systems designers at any major commercial software company to describe the architecture of their product on a whiteboard. They'll typically draw a diagram showing a number of boxes (modules) that perform highly specific functions, with a few neat connections between them. My research shows however, that if you actually measure the interactions between boxes *at the code-level*, you'll find the architecture is much more tightly coupled than anyone would think. Coupling has its virtues—tight interactions between different pieces of code can lead to increased performance in areas such as speed or memory footprint. But coupling also has major drawbacks, with respect to the ease with which software can be corrected and adapted to meet future needs.

Virtual systems are fundamentally different from other kinds of systems. As an information-based product, software appears to be easy and quick to change—which can be an advantage and a disadvantage. There are no physical changes to be made, yet the complexity of modern software is such that even small modifications can ripple through a system with unintended consequences. Software appears to be malleable, but in practice, the architecture of many systems is opaque. A developer dare not change them too much for fear of creating a tangled web of dependencies and changes to upstream files.

Furthermore, unlike industries such as automobiles and airplanes, which create new platforms from the ground up every few years, modern software development efforts rarely start with a clean slate. Most systems have a significant legacy, on top of which new features and functionality are built. Unfortunately, it's not obvious from looking at the older code which pieces are connected to which others. It's not like working with a mechanical system, where you can see connections simply by inspecting the product, or reverse-engineering its design. Unfortunately, this hard-to-understand legacy code often embeds assumptions and design decisions that are no longer optimal for the system.

Why are initial design decisions often so out-of-whack with the current requirements for a software system? One reason is that the original design may have been built quickly, by a small company or startup more focused on releasing its first product rapidly than on building a framework to last for many years and multiple product evolutions. Software engineers design programs to meet their immediate needs, and in a startup, there is no guarantee that you will be around in 12 months. Speed is of the essence, and any performance edge is pivotal, no matter how you achieve it. Ten years later, however, when the war for market share is over, the needs of a user might be better served by a much more modular, maintainable, and adaptable system. In essence, early design decisions create a "technical debt" that must be paid by all those that follow.

Let me provide a micro-level example of these dynamics. Alice might decide to use a piece of functionality that Robert has already designed in his module, so she writes some code to "call" his function from her module. This saves time, but creates a dependency between Alice's modules and Robert's that may not be transparent to the system architect. Five years down the line, when Robert and Alice have both retired to Tenerife, that dependency may be a complete surprise to a programmer needing to make a change. Changing code in Robert's module may well cause Alice's module to cease functioning.

The work that Andrei Akaikine, SDM '09, did in the thesis II supervised provides a great example of the costs that arise from an architecture that is overly complex. In his thesis, he examined a software system with a long history, which generated significant maintenance costs each year. Every change could create unexpected problems and require additional fixes to other parts of the system. The owner of this system—a large commercial software firm—decided to redesign the software with the goal of adding new features to the system, while simultaneously reducing its complexity (by reducing the coupling between elements). Akaikine showed that the result of this redesign was a significant reduction in maintenance effort, as captured by the time it takes to fix defects.

Of course, any major redesign involves significant costs of its own—management has to decide if these costs are warranted. Unfortunately, many businesses make these decisions based on gut-feel and intuition, rather than a rigorous analysis of the likely payoffs. We need much better data to make informed decisions, and the software industry is woefully lacking in such data. Ultimately, this is why the work I have done with Akaikine and other ESD students—including Daniel Sturtevant, SDM '07, who is working on his PhD—is important. We are among the first research teams to visualize and measure the extent of technical debt in legacy software systems.

To achieve this goal, we have developed pioneering methods for visualizing and measuring attributes of a software architecture that can help us assess its underlying structure. Consider a well-known example from a recent paper, in which we look at the Mozilla web browser. After its release as open-source software in 1998, a major redesign effort was undertaken on the system, with the aim of making the codebase more modular, and hence easier to contribute to. The design structure matrices (DSMs) from before and after this redesign (see Figure 1) illustrate what happened. The modular architecture that resulted facilitated contributions to the code by creating fewer unintended interactions between components. Before the redesign, each component was, on average, connected to 18 percent of other components. Afterward, this figure dropped to below 3 percent.

Ultimately, different designs will have different performance characteristics along a variety of important dimensions, making techniques like ours valuable for exploring design trade-offs. A highly integrated design is likely to be faster, while a highly modular design may be more reliable. A designer must consider carefully what the product needs to do to arrive at the optimal design for her objectives. For example, if a system has to last 10 years, and you have no idea what it will need to do at the end of that time, the software must be designed to be extremely flexible and evolvable. Unfortunately, very few software companies practice such forward-looking "systems thinking."

How should a firm begin? Nobody should rush headlong into full-blown re-factoring of a major system, given we are still in the infancy of understanding how these efforts work. Indeed, our research reveals that a manager's intuition about where to start such an effort is frequently wrong, given the perceptions of an architecture and the realities embedded in its source code are often in conflict. Software companies first need to generate data on measures of architecture, and begin to link these measures to performance outcomes that they care about. Most firms tinker with and redesign their software all the time—in effect they run hundreds of small experiments every year. Armed with a careful assessment of this data, they will be better placed to assess what works and what doesn't. Ultimately, we know complexity hurts. But reducing it is also a complex endeavor.

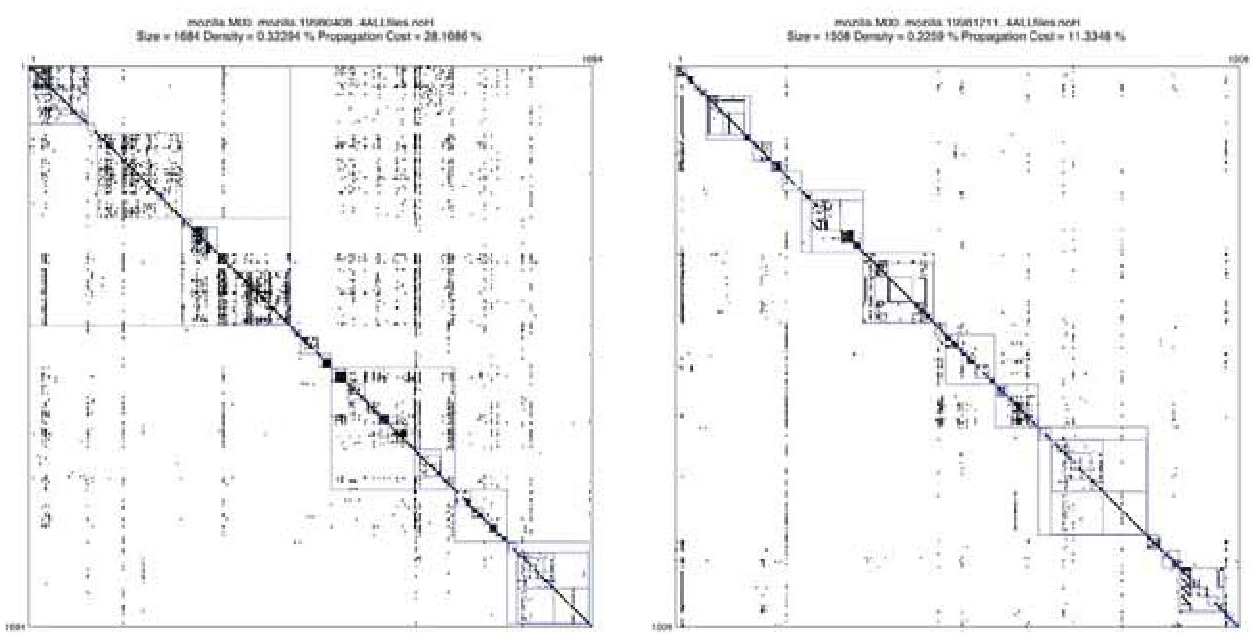


Figure 1. These two design structure matrices illustrate the interdependencies that existed within Mozilla's software architecture before (left) and after a major redesign.

SDM alum explains need for negotiation in software architecture

By Christine Miyachi, SDM '00

Christine Miyachi, SDM '00

Editor's note: *SDM alumna Christine Miyachi is a principle systems engineer and architect at Xerox Corporation. She also writes a weekly blog about software architecture: abstractsoftware.blogspot.com.*

"An accidental architecture emerges from the multitude of individual design decisions that occur during development, only after which can we name that architecture."

—Grady Booch, pioneering software engineer

I enrolled in MIT's System Design and Management Program (SDM) because I wanted to become a lead software designer/architect/systems engineer. I thought the program would give me the technical skills I needed—and it did. But I also learned that leadership and negotiation are critical to architecting successful software.

Today, my daily work involves myriad decisions that depend on both technical expertise and an understanding of business needs. My background gives me the skills to create an initial framework that will persist. However, even after an architecture has been designed and implemented, information about how the system will and can work is often discovered, requiring design shifts to continue to support the business.

It's important to understand—and to be able to communicate—that creating software is not like an assembly line where every piece of work is completely uniform and understood. In fact, it's an intellectually intensive and often unpredictable process.

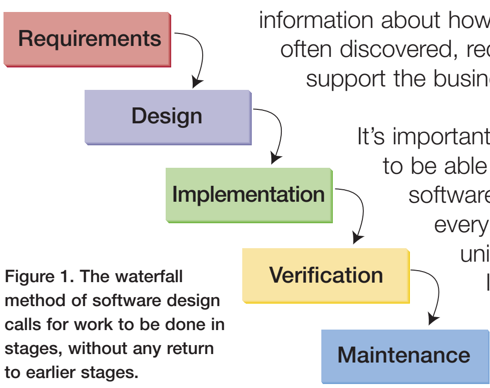


Figure 1. The waterfall method of software design calls for work to be done in stages, without any return to earlier stages.

It's helpful to compare and contrast two methodologies software developers commonly use: waterfall (Figure 1) and iterative/agile (Figure 2). Pure waterfall proponents believe that all architectural decision must be made upfront. Pure agile proponents say that nothing needs to be decided upfront because methods, requirements, design, and implementation is done in short cycles, with functionality delivered to the stakeholders in each iteration. However, most software architects agree that a combination of upfront and iterative decisions contribute to successful software architecture.

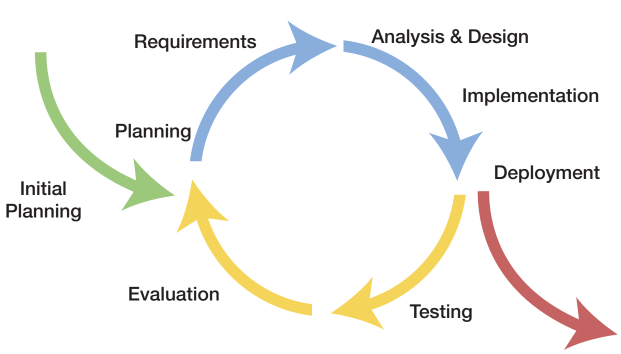


Figure 2. In iterative or agile design, requirements, design, and implementation is done in each short cycle, with functionality delivered to stakeholders after each iteration.

In Ken Schwaber's book *Agile Software Development with SCRUM*, he discusses the empirical model of a process that incorporates the unexpected. Control is established through frequent adaptations, he explains. My experience with software architecture is very close to this. While it is tempting to be very rigid in architecture and not allow changes, this strategy causes conflicts and often produces a system that does not meet the ever-changing business needs.

For example, in one project I worked on many years ago, we selected a framework that auto-generated code. This framework became outdated, the company that provided the auto-generating tools went out of business, and we got to the point where we couldn't make changes to this module in our system. We actually had to design other programs around it. Finally, thanks to the courage of a few, this section was redone.

The challenge in software is that, unlike other engineering disciplines, the output is malleable. Changes can be made fairly easily—at least before the product ships. I think that's why I've found we always end up with what Grady Booch calls an "accidental architecture." The final product is a result of decisions made following each iteration—and all of these decisions require a lot of negotiation.

Here are three examples of how design and architecture are negotiated in software. They illustrate the tools and skills I learned while an SDM student, in particular the leadership and negotiation methodologies that are critical to architecting successful software.

Case 1: I'm not the expert, but I bring a team of experts in to make a decision.

Experts from different disciplines often don't agree, so in this case I act as a negotiator, guiding the team to a solution—usually in one meeting because time is always a factor. When negotiating and brainstorming, I make sure that everyone's voice is heard, and I work to get the quiet people to speak up. It's important not to make anyone appear stupid, and not to use anger or aggression to win your case. That usually backfires.

I use tools such as mind maps or decision matrices to

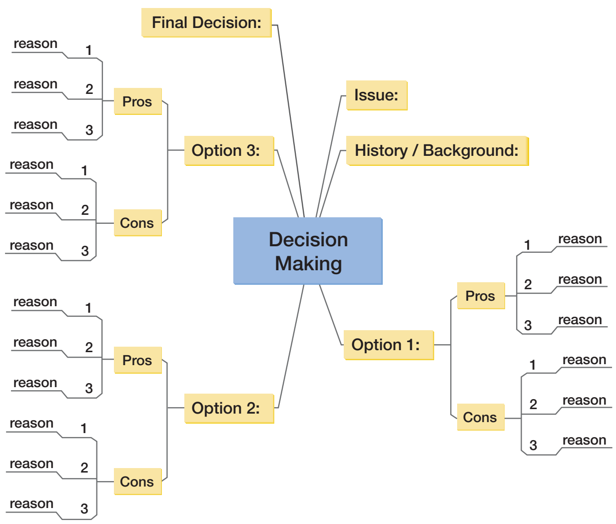


Figure 3. Using a template like this one to establish a mind map can help bring logic and rigor to the decision-making process.

bring logic, rigor, and a common understanding to the discussion. The mind map (Figure 3, a tool taught in SDM's system architecture course) and the decision matrix (similar to the quality function development matrix taught in SDM's product design class) are both ways of visually representing the relationship of ideas. Axiomatic design, another method taught in the SDM program, is similar.

For more information about these tools, visit http://abstractsoftware.blogspot.com/2010/08/architectural-decisions-or-on-purpose.html.

Case 2: I disagree with someone else on a design decision.

When should I dig in my heels and when should I compromise?

I have a colleague on my team who always wins design arguments because he has a very strong personality. Other people back down simply because they don't want to put up a fight. Sometimes I do fight, but I can also compromise. The question is when to hold my ground and when to give way.

The following guidelines, taught in SDM, have proved useful:

- If I realize the other person is correct (I see something I didn't see before), I will agree to change.
- If I think the proposed decision direction will hurt the project, I will fight it.
- If I think we can make a change later with no ill effects, I will sometimes wait and put the change in a parking lot.

Of course, I'm not always right. Sometimes I end up wishing I had fought harder and sometimes I wish I had compromised. I'm always learning.

Case 3: A key decision is made, then changed.

This scenario is common in software architecture, partly because of the product's malleability and partly because leadership and teams are subject to change.

For example, I once acted as a negotiator and an expert for a key architectural decision. We developed a set of requirements, weighed their importance, and looked at four solutions to meet those requirements. We rated each solution against those requirements, and two were very close. One was more robust but not as scalable to low-end computing systems (not a lot of memory, weaker CPU, etc). One was less robust but very scalable. Although one team member objected, the team ended up choosing the more robust system.

About a year later, after some shifts in the business, we had to revisit the decision. The market had changed and we had to go with a less robust but more scalable solution. The lesson is that as an architect you must be able to gauge if a solution is still viable in the business climate, which is constantly and sometimes drastically changing. We had to adapt or we would have lost business. It's also essential to document decisions as they are made because this will save time and increase efficiency by eliminating the need for "archaeological excavation." In short, if the business changes, you will more easily see when past logic is no longer valid and know that it's time to make some changes.

The bottom line is that software architects need more than technical and managerial expertise—they need to know how to negotiate design decisions and use a toolbox of ways to examine tradeoffs logically—something SDM provided for me.

Today, these SDM tools help enhance the contributions I and my team members can make to Xerox. This includes the expertise to facilitate and lead groups to create an initial framework that will persist, but as importantly, the leadership and negotiation skills to make design shifts that will continue to support the company as it evolves.

SDM Best Thesis analyzes costs of software system complexity

> continued from page 1

based on data from the industry. The core idea was to demonstrate a correlation between some numerical measure of a system's complexity and engineers' productivity.

I studied two versions of a mature software product. The chosen product had undergone a significant redesign between these two versions that involved rewriting an estimated 70 percent of the product's code. However, most of product's functional requirements did not change. This provided a basis for an experiment with few externalities.

In the experiment, I measured the effort engineers spent making a code modification associated with a corrective change request. For each version of the product, I tallied about 400 corrective changes performed during the first 30 months after release. I then used the time that engineers reported working on each change to estimate their productivity.

To measure the initial product architecture complexity of the product, I chose the methodology developed by Alan MacCormack, MIT visiting associate professor of technological innovation, entrepreneurship, and strategic management (see article, page 4), and Carliss Baldwin, the William L. White professor of business administration at Harvard Business School. This methodology is based on the application of design structure matrices (DSMs) to map dependencies between system components. I used DSMs to compute the design structure complexity measures that were most suitable to assessing software system maintainability (see Figure 1).

The first complexity metric I used measures the degree of "ripple effect" propagation through the chain of dependencies across system elements. Propagation cost predicts the percentage of system components that can be affected, on average, when a change is made to a random design element. Propagation cost is a single numerical measure that characterizes the interconnectedness of the elements that comprise the product system.

For each design element, two other complexity measures can be defined: 1) a measure of dependencies that flow into an element—fan-in visibility; and 2) a measure of dependencies that flow out of the element—fan-out visibility. Based on fan-in visibility and fan-out visibility measures, components of the system may be characterized into four types. MacCormack et al. define these types as core, peripheral, shared, and control (see Figure 2). These few measures captured the important complexity attributes of the hypotheses I tested in my research.

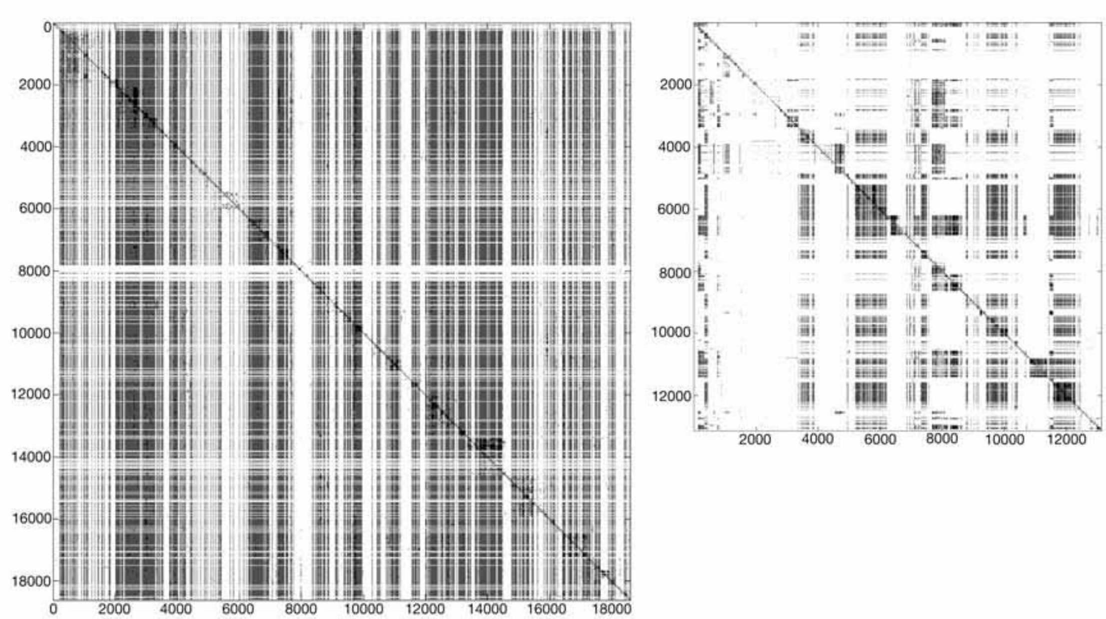


Figure 1: Visibility matrices of the product before and after the redesign. Visibility matrix for the old product is much denser, indicating the high propagation cost.

My experiment showed a clear correlation between initial product complexity and engineers' productivity. A decrease in the product's propagation cost (or complexity) from 38 percent to 11 percent resulted in 10 percent to 14 percent improvement in engineers' productivity. Besides engineer productivity, such metrics as amount of rework and effort needed for a single module change improved by 16 percent to 25 percent.

	Fan-In Low	Fan-In High
Fan-Out Low	**Peripheral Component**	Shared Component
Fan-Out High	Control Component	Core Component

Figure 2: Characterization of components by visibility measures.

Modules of the core type were found to be more likely to be touched during a random corrective code change. Out of 542 unique modules that were changed, 404 were of the core type. The frequency with which core modules were modified is disproportionate to their share in the whole system: 85 percent of modules that were modified more than once during the studied period were core modules. Some of them were modified 10-12 times.

I concluded that product architecture redesign that reduces the structural complexity of the software system may become economically viable and desirable as the product support cycle grows in length and the pressure to lower maintenance costs increases. Considering that in some cases, maintenance can consume up to 90 percent of the total product life-cycle cost, an improvement of 10 percent to 15 percent in productivity may result in savings almost as large as the initial cost of product development. Of course, any redesign may have downsides that need to be considered before the decision to redesign an existing product can be made. It is essential for a development organization to perform a comprehensive net present value (NPV) analysis before investing resources into redesign to avoid any NPV-negative initiatives.

That said, it's clear that the economic benefits of controlling the complexity of product architecture have significant managerial implications. Product development organizations should measure the complexity of new and legacy software products to control their total life-cycle costs.

SDM course helps students build a leadership roadmap

> continued from page 3

bringing people to their own solutions—an effort that builds trust and loyalty. Leaders need to know how to build and sustain trust in order to motivate followers. And, once trust breaks down, it can be impossible to regain. For that reason, the course also includes simulated conflicts that allows students to experience the corrosive and contagious nature of mistrust.

As a final exercise, each student is asked to submit a paper analyzing an unsuccessful experience. This can range from having a conflict with a boss to not getting along with a peer. By reflecting on the experience and relying on the findings from the various instruments,

> As leaders, we will have to know how to get the best out of people no matter what their personalities. We learned how to lead and structure work for individuals with a variety of personality types, building concepts and developing practices that will be effective among as diverse a group as possible.
>
> *—Matt Harper, SDM '10*

students are able to understand the dynamics and their roles in making things worse. By examining their role, they come up with constructive alternatives to the problem they faced.

Lastly, each student has the option of meeting the instructor for one hour privately to go deeper into the results of the instruments and to get coaching and counseling. By the end of this course, students have learned to enhance their effectiveness as leaders—developing a set of "soft skills" that often make the crucial difference in human relations.

SDM alumni entrepreneurs use systems thinking to drive success

***Editor's note:** The curriculum of the MIT System Design and Management Program (SDM) integrates coursework and lectures on management, technology, and social sciences to prepare graduates to innovate in, and lead, organizations. Here is a look at five SDM alumni who have used systems thinking to start and grow new businesses.*

Solar-powered refrigeration

Sorin Grama, SDM '06
Cofounder and CEO
Promethean Power Systems, Founded 2007

I joined SDM because I wanted to learn more about business and engineering issues affecting larger organizations. Ironically, I changed paths and decided to start my own business, but SDM gave me the crucial understanding of how large organizations operate—insight that is often necessary to run smaller, entrepreneurial organizations.

Sorin Grama, SDM '06, cofounder and CEO of Promethean Power Systems, with his cofounder, Sam White.

My partner and I founded Promethean Power Systems in 2007, right after I graduated from SDM. We had traveled to India to evaluate one possible business opportunity, and discovered a different one in the Indian dairy market. India is the largest consumer and producer of milk in the world, but most milk comes from remote villages that lack refrigeration. As a result, raw milk must be collected twice a day. Transportation costs are high, and a large percentage of milk spoils before it reaches the processing plants. We formed Promethean Power to develop a hybrid solar-and-grid-powered milk chiller that can be operated wherever the electric grid is unreliable or nonexistent. The systems are sold directly to dairy plants that have the need and buying power to purchase these units.

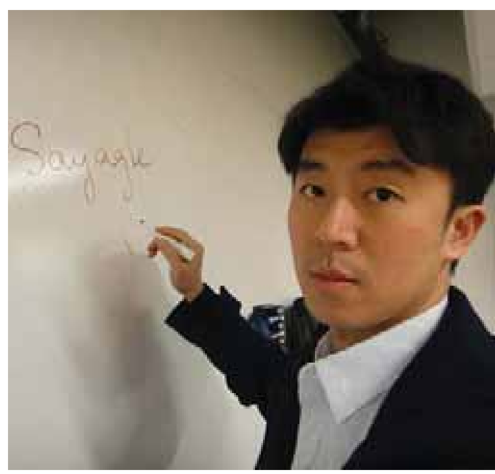
My SDM training was instrumental in starting this company. SDM's course in product design and development underscored the importance of first understanding the customers' needs and then developing the product to match that need. System architecture, project management, and system dynamics have also helped us to craft the initial product architecture and form our team. To date, we have developed three prototypes and performed an extensive field trial in India. Our latest generation product was shipped to our first customer in India at the end of January.

Going forward we plan to transfer the technology to our Indian subsidiary and establish an initial production facility in India to meet anticipated customer demand

Web services

Ken Huang, SDM '05
President, Founder, and CEO
Sayagle, Inc., Founded January 2009

Sayagle is a location-based social marketplace designed to be a win-win-win arrangement for users, merchants, and charity organizations. Users can make the most of their day-to-day transactions with Sayagle's local deals and recurring rewards program. And merchants—particularly brick-and-mortar stores that have not been able to capitalize on electronic marketing—get access to promotional tools that drive local traffic in real time. Sayagle has also created a new charity donation



Ken Huang, SDM '05, is the president, founder, and CEO of Sayagle, Inc.

method—partner charities can collect receipts from Sayagle's merchant network instead of asking people for cash donations.

Founded in January 2009, Sayagle had its soft launch in September 2010, establishing the time-stamp for my two US patent applications. The official launch should occur before this summer. The company has 25 employees, and more than 250 Boston-area businesses have joined Sayagle to offer user deals.

The SDM program has helped me to identify key variables that are on the critical path for my business, as well as the cost-benefit analysis I apply to every aspect of my company. Most importantly, I am taking a holistic view to project the company direction along with the business strategies.

My goal going forward is to roll out Sayagle as a fully bundled package to the local community and continue to recruit top-notch talent to make Sayagle successful. Hopefully, I will create a mutually beneficial equilibrium in the Boston community and contribute value to society. Sayagle's vision is to enhance life beyond the screen.

Internet marketing

Yoav Shapira, SDM '05
Vice President for Platform Strategy
HubSpot, Founded 2006

HubSpot is doing exceptionally, amazingly well. We help small businesses with their marketing using a new methodology—inbound marketing. Old methods like direct mail and other strategies that are based on interrupting customers either don't work or are diminishing in effectiveness. Our philosophy is that if you do some research and distribute content that people want, customers will find your business in the course of their activities.

Yoav Shapira, SDM '05

The company started in June 2006 right after the two founders graduated from MIT. I knew them at Sloan, and they approached me to start the company with them, but I went instead to another startup first and joined HubSpot a year later. Today HubSpot has expanded to about 4,000 paying customers. Just last month we added almost 300 more. Close to 200 people work for the company. It's growing very fast and we've successfully raised venture capital three times.

SDM has definitely helped me in my business. For example, we frequently do the sort of modeling taught in Professor David Simchi-Levi's system optimization course, and I had no background in that area prior to studying at SDM. I use accounting every month. And, I found the course in disruptive technologies taught by Professor James Utterback just transformative. We discuss the theory at management meetings, asking ourselves questions like "who do we disrupt?" and "how do we keep others from disrupting us?" That was probably my favorite class at MIT.

Speech indexing software

Ben Jiang, SDM '08
Cofounder and CEO
Nexiwave, Founded 2008

Shortly after joining SDM, I became obsessed with the path that the information technology (IT) industry was taking in helping people find the information they need. One area that had been less explored by the IT industry was content in the audio format. While people communicate vast amounts of information by speech, it was virtually impossible to find and extract information from audio files.

My partners, including Cynthia Munoz, SDM '08, and I founded Nexiwave to search audio files for information—a service that

Ben Jiang, SDM '08, cofounded Nexiwave to make finding information in audio files easier.

> continued on page 12

SDM alumni entrepreneurs use systems thinking to drive success

> continued from page 11

was not widely accessible before. Nexiwave shipped its first product, the core speech indexing and audio search feature, in December 2008. In February 2009, Nexiwave sponsored and developed a conference call service, known as SearchMyMeetings. SearchMyMeetings is a demonstration platform of the benefits of our audio search feature. It processes audio recordings for fast keyword search and playback. SearchMyMeetings has signed up the MIT Sloan School of Management as its first major user.

From the beginning, SDM has been a big part of our success. SDM concepts of marketing and industry analysis have helped us to hone our products, and system project management has enabled us to determine what steps to take along the way. Accounting has proved invaluable, as I now understand financial matters better and can speak knowledgeably with our finance person. Finally, SDM students were major users during the company's infancy.

Moving forward, Nexiwave expects to utilize its knowledge to broaden its customer base, as well as the company's focus.

Electrical power systems

David Sharman, SDM '01
Managing Director
Ampair, 2003

Ampair manufactures small-scale renewable and distributed electrical power systems. Our ambitions are in the sub-100kW range, and most of our existing work is in the sub-10kW range. The company currently manufactures wind turbines, hydro turbines, and phase converters. We also do work in adjacent spaces that we expect will grow.

David Sharman, SDM '01

The business really began in 2003 when I took over an existing phase converter company that employed about half a person (which became me—at the beginning I did everything from design to assembly, installation, and accounts). Then, in 2005, we acquired the Ampair wind turbine business, and now we have grown to about 15 people (we are recruiting more—good mechanical engineers are welcome right now). We generally spend over 25 percent of turnover on R&D, and we sell 1,000 systems a year in about 50 countries. Our clients range from individuals to major corporations. Last year we took in our first round of venture capital, which has allowed us to invest and make faster progress.

The Ampair 6000 wind turbine is shown at a test site in Berkshire, UK. David Sharman, SDM '01, is the managing director of Ampair

For me, the SDM experience was an intense period of learning in concert with many high-quality people who understand systems thinking. That experience helped me to identify a viable space within which to build a business and then to create a valid framework for not missing anything that is mission-critical at a holistic level.

Now I am keen to grow the business to the point where it is robust in all economic circumstances and can withstand the loss of any key individual, including me. That will give me the satisfaction of knowing that the livelihoods of our people are assured.

This report was compiled by Kathryn O'Neill, managing editor, SDM Pulse.

MIT SEAri and TUM collaborate on research on uncertainties in product platforms

By Augustin Friedel, visiting research student, SEAri

Augustin Friedel

Systems Engineering Advancement Research Initiative

Editor's note: *Augustin Friedel is a master's student in mechanical engineering and management at Germany's Technische Universität München (TUM.)*

For the last six months, I've worked on a research project investigating uncertainties in product platforms at MIT's Systems Engineering Advancement Research Initiative (SEAri). The research was guided by SEAri principal research scientist Dr. Donna H. Rhodes, and TUM Aerospace Professor Eduard Igenbergs.

I saw the huge potential of the platform approach in complex systems while working on a platform project for a client of a consulting company that specializes in automotive, transportation, and aerospace systems engineering projects. I also became aware that the process of platform design was, in general, not explicitly understood among members of teams working on a common problem.

I realized that some of the confusion and uncertainty around the platform development process could be reduced through structured methods. This would allow teams to develop successful product platforms more efficiently and turn their companies into market winners. Complex systems, including product platforms, are subject to uncertainties that may lead to suboptimal functional performance or even catastrophic failures if unmanaged over time. Identifying uncertainties in the front-end—and implementing ways to mitigate problems that may occur—can be a part of the product platform design process that adds value to the platform as a system.

My thesis describes the journey of developing a method for investigating the management of uncertainty that will help platform developers with the complicated process of transforming an idea into a finished product. In it, a platform is defined as a set of architecture, common modules, and interfaces from which a stream of derivative products can be efficiently developed and launched. The architecture is the configuration within the product; a module is a part or a group that allocates a function to the product; and the interfaces are connections between the modules and architecture, among the modules themselves, and between the platform and customized parts of the product.

The thesis has three sections. In the first, I discuss the results of an empirical case with 10 participants (on hierarchy levels between lead engineer and senior project director in automotive, electronics, agriculture machinery, and defense industries), studying the uncertainties in product platforms and the consequences. This study found that most uncertainty occurs for reasons that could have been predicted. Furthermore, companies often realize too late that there is a problem because of an uncertainty, reacting only after discovering that the platform didn't meet expected performance criteria.

Given the findings, there appeared to be a need for an approach to managing uncertainty that would allow a value-robust platform to be built. In the second part of the thesis, I describe developing a framework for that purpose.

The preliminary framework has seven steps, providing a tool for managing uncertainties and risks within a platform (see Figure 1). The first two steps are based on the Quality Function Deployment method of transforming customer requirements into engineering metrics. The third step is based on the ISO standards for managing risk by

> continued on page 14

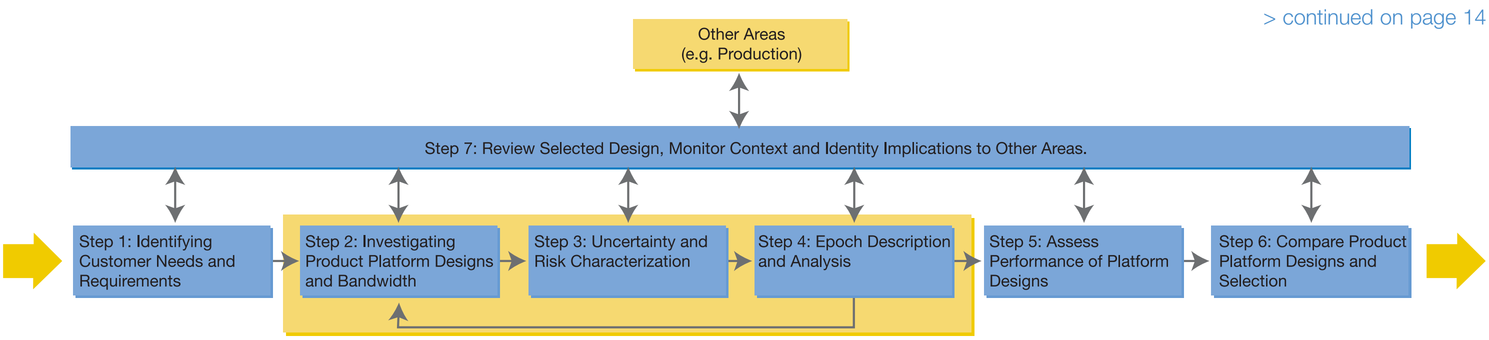


Figure 1. Overview of the preliminary framework for managing uncertainties in product platform lifecycles.

Employers recognize value of hiring SDM graduates

By Helen M. Trimble, director, SDM Career Development

Helen M. Trimble

Interest in SDM graduates continues to accelerate as more companies become aware of the value of hiring MIT graduates who have both technical and management expertise. Recent SDM Fellows have assumed top-level leadership positions across a wide range of industries.

SDM takes pride in educating its students to assume leadership and technology consulting roles in product development, business strategy, and operations for industries and organizations as diverse as nonprofits, aerospace, health care, pharmaceuticals, and financial services. Examples of positions into which our graduates were hired this year include manager of business development, senior systems engineer, vice president of engineering, founding manager, and director of technology strategy.

Following is a brief overview of the placement statistics for the 2009-2010 year. Although many students come to SDM through company sponsorship, for those who do not, SDM compiles the data and publishes an employment report each year. This report includes 22 self-funded SDM students hired by world-class organizations. During the recent economic downturn, these organizations have recognized the importance of hiring people with the right skill set to be competitive into the future.

Highlights of the employment survey results include:

- 100 percent of SDM Fellows who responded to the survey are employed, with 90 percent accepting employment before graduation in May 2010, and the remaining 10 percent accepting employment within one month of graduation.
- SDM Fellows received an average base salary of $105,037 (median $105,000), which represents an increase of $28,187 or 37 percent over their average base salary prior to entering the SDM program.
- Product development/management job functions drew 25 percent of the 2010 graduates followed by consulting/strategy at 20 percent.
- 55 percent of the SDM graduates will be eligible for a guaranteed annual bonus or performance bonus averaging 16 percent of salary

We invite your company to participate in upcoming SDM recruitment activities and to interview our self-funded SDM students. Contact Helen Trimble at htrimble@mit.edu with any questions or for further information and a copy of the 2009-2010 SDM Employment Report.

MIT SEAri and TUM collaborate on research

> continued from page 13

identifying, analyzing, and evaluating different uncertainties. The fifth step describes an approach for treating the uncertainties and risks of implementing mitigation mechanisms in critical parts of the platform.

Critical parts are identified on two different paths in Step 4 of the framework. The first starts with tracing the impact of uncertainties on the components; the second path traces the impact of each uncertainty on the engineering metrics. (For predicting an instance where exercise of a mechanism is needed, the Epoch-Era Analysis developed by SEAri Research Scientist Adam Ross and Rhodes was adapted.) Step 6 compares platform designs and selects the most valuable one with a Multi-Attribute Tradespace Exploration methodology, developed by Ross. Step 7 runs parallel to the other steps, and the monitoring aspect allows reaction on uncertainty before the risk can take effect.

The third section of the thesis successfully applies the framework to the example of a a platform-based cleaning robot.

Future research can serve to evolve and further test this framework. The final thesis, “Investigating the Management of Uncertainty in Product Platform Lifecycles,” as well as more information on SEAri and its research, is available at seari.mit.edu.

SDM welcomes talented, diverse 2011 cohort

By Kathryn ONeill, managing editor, *SDM Pulse*

This January, MIT's System Design and Management Program (SDM) welcomed more than 70 mid-career professionals into its 2011 cohort. Class members hail from all over the globe—from Spain to Saudi Arabia and from the United States to China—but share a common goal: to learn to lead effectively by using systems thinking to solve large-scale, complex challenges.

The SDM 2011 cohort poses with SDM Fellows Program Director Pat Hale, back row, sixth from right.

"After five years in purely engineering roles, I was looking at programs that would enhance my managerial and leadership perspectives while at the same time leveraging my technical background. I found SDM to be the best fit," said Farrah Tazyeen, SDM '11, who came to SDM from a position in product development at Oracle Solution Services India. "I am discovering that, as an SDM Fellow, all the amazing resources at MIT are within my reach."

The cohort Tazyeen joined is impressive. Several members of the class have already earned at least one master's degree in disciplines ranging from software engineering to physics and finance. A few have MBAs and even PhDs.

But the numbers don't tell the whole story. Victor Piper, SDM '11, who works as a quality assurance engineer at Raytheon, said that meeting SDM students at an information session is what convinced him that the program was the right choice. "What stood out was the way the students spoke about how the program changed their way of thinking," Piper said.

Piper already has a master's degree in electrical engineering, but he wanted to build a skill set that would help him find and solve technical problems with broader enterprise impact. "[SDM] seemed unlike other engineering management programs, as it is in equal measures rigorously technical and enterprise-focused," he said.

But SDM's academic program is not its only asset. The SDM '11 class—like its predecessors—includes a diverse mix of people. There are experts in such disciplines as communications engineering, mathematical statistics, biosystems engineering, and management, as well as one member who holds a bachelor's degree in Oriental studies and another whose master's is in philosophy. The classmates hail from well over a dozen countries.

That diversity appealed to Melissa Rosen, SDM '11, who works as an engineering consultant in the medical devices industry. "After 10 years of hands-on experience in my field, I am ready to be exposed to other industries and disciplines," she said. "MIT is a technological hub that attracts many companies and leaders of industry; as a SDM student, this network is invaluable for career opportunities."

Rosen also said she was impressed by how well the program accommodates those who can't afford to be full-time, on-campus students. "Not only does SDM provide the flexibility of being 'career compatible,' but it provides the necessary tools to advance immediately at my current position without having to take a break from industry."

The new cohort began in January with an intensive

> continued on page 16

SDM welcomes talented, diverse 2011 cohort

> continued from page 15

month on campus commonly called “SDM boot camp.” Designed to help class members to bond, the January program includes instruction in system architecture, a leadership course, several demanding team design challenges, and the popular Human Side of Technology class.

“In the January session I was exposed to principles of system architecture, team dynamics, and organizational processes. I worked more than I thought myself capable and bonded closely with my cohort,” Piper said. “It’s already been an incredible experience, so I find myself very much excited for all that’s to come.”

The January session was just the first step for the SDM ’11 class, which will go on to complete 13 to 24 months of coursework in system architecture, systems engineering, and system and project management, as well as a master’s thesis.

“The journey so far has already been transformative,” Tazyeen said. “There’s lots to learn and I’m looking forward to exciting challenges ahead!”

One long-term benefit will be the connections that SDM provides. “Within a short amount of time, the feeling of the ‘SDM family’ at MIT emerged,” Rosen said. “It is clear that this is just the start of a life-long network with incredible individuals.”

SDM 2011: A typical SDM class

A quick glance at the titles held by members of this year's cohort and the companies from which they hail provides a glimpse into the depth and breadth of experience represented in every SDM class.

Chief Evangelist, Skype

Lead Software Engineer, Analog Devices Inc.

Industrial Systems Engineer, Saudi Aramco

Engineering Manager, GE Energy

Product Engineer, Ford Motor Company

Scientist, Boston Scientific Corporation

Creative and Managing Director, Cofounder, Skyrill

IT Manager, Texas Instruments

Project Engineer, Helbling Precision Engineering Inc.

Quality Assurance Engineer, Raytheon-Integrated Defense Systems

Senior Architect, Philips

Senior Business Analyst, Fidelity Investments

Senior Engineer, Procter & Gamble Co.

Senior Product Engineer, Deere and Company

Software Architect, Xerox Corporation

Software Engineer, Cisco Systems Inc.

Staff Software Engineer, IBM

Strategy Engineer, LAN Airlines

Sub-region Lead Account Manager, Nokia Siemens Networks

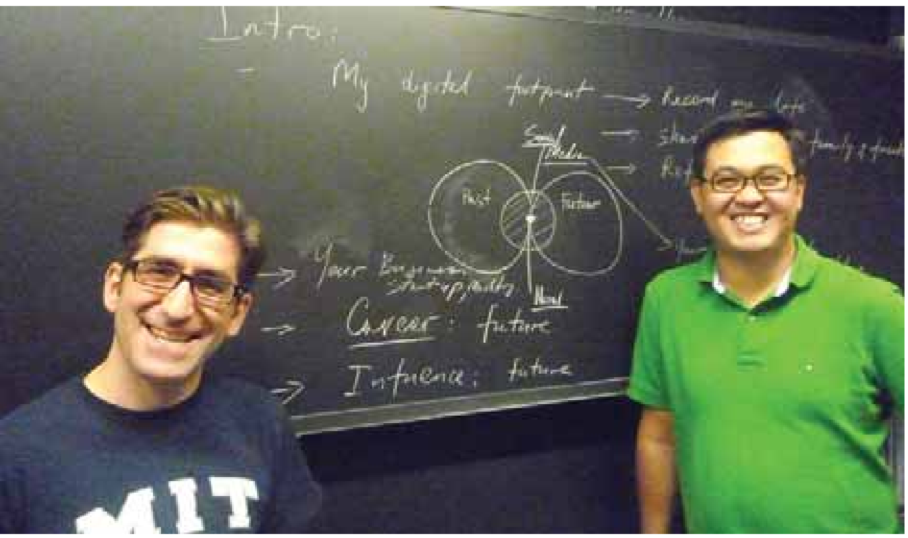
System Engineer, Ansaldo Railway System

Vice President, Tonghe Architectural Design & Urban Planning Co., Ltd.

SDMs launch MIT Social Media Club

By Rafael Marañón-Abreu, SDM ’10, and Azamat Abdymomunov, SDM ’10

Editor’s note: *Rafael Marañón-Abreu and Azamat Abdymomunov founded the MIT Social Media Club in September 2010. As of this writing, the club has 70 active members, including MIT students, faculty, and staff.*



Two members of SDM’s 2010 cohort, Rafael Marañón-Abreu, left, and Azamat Abdymomunov, teamed up to start the MIT Social Media Club.

MIT’s System Design and Management Program (SDM) has not only provided the advantages that we expected from a world-class program in engineering and management—it has also given us the opportunity to learn about emerging technologies that can help people, corporations, and government work more effectively. Within this realm, social media stood out for us as an area worth further exploration.

We cofounded the MIT Social Media Club to build connections with others at SDM and across the Institute and work collectively to understand the newest channels of communication—from communities such as Facebook and LinkedIn to blogs, Twitter feeds, and YouTube. As mid-career professionals returning to university, we believed that it was important to investigate social media and understand how to put them to work for individuals and organizations. We were surprised to learn that there was no social media club at MIT, so we decided to start one. We believed this could help us maximize our education and share past and present thoughts and experiences, while visualizing and creating our individual futures and simultaneously giving back to SDM and the MIT communities.

Founding the MIT Social Media Club was hard work, but applying many of the concepts we learned in our SDM courses helped us to execute this exciting startup. For example, our lessons from classes in system architecture, systems thinking, technology strategy, and project management helped us to look at how social media functions in the contemporary environment.

We discovered that social media is not only useful for job hunting, but can help us better understand the dynamics among talented people in an organization, as well as how learning teams are constituted and flourish. We believe it can offer a competitive advantage in global business, help governments reinvent themselves, and help academics expand and evolve their capacity for teaching and research.

In the MIT Social Media Club, we encourage our members—including PhD, master’s, and Sloan students, as well as others at MIT—to understand and get hands-on experience using social media tools and to explore how they can be used to close the gap between an organization’s senior leaders, front-line employees, partner companies, customers, and other stakeholders. In the same way, social media can be used to build bridges between faculty members and students, and among researchers from different universities and countries.

Already we have come up with a couple of frameworks that we used in teaching an Independent Activities Period course this past January, ESD.942 Social Media: Trust, Information Seeking & Systems Innovation in the Digital Age. This class was sponsored by Dr. Joseph Coughlin, director of the MIT AgeLab, who led the first multi-disciplinary research program created to understand the behavior of the 45+ population, including how the older population makes decisions using social media.

Although still very new, the MIT Social Media Club has held several successful events, including workshops on how to increase your digital footprint and how to use social media in a job search. This spring we’re planning a series of social media research tours, which will allow club members to visit different departments and labs at MIT and elsewhere to explore how social media are being used in the workplace.

We would like to extend an invitation to SDM industry partners to get involved. For more information, visit the MIT Social Media Club online at socialmedia.mit.edu.

Keynote speakers announced for systems thinking conference

By Lois Slavin, SDM communications director

On October 24-25, 2011, MIT's annual systems thinking conference will be held at Wong Auditorium on the MIT campus. Sponsored by MIT's System Design and Management Program (SDM), the conference focuses on using systems thinking to address today's complex challenges and achieve success.

Ed Crawley

Julian Goldman

MIT professors will frame the three-fold nature of systems thinking—technical, managerial, and socio-political—and outline how it can be applied. Industry leaders will describe best practices that demonstrate the challenges they face within and outside their organizations, how they use systems-based approaches, the benefits achieved, and the lessons learned.

MIT SDM is pleased to announce this year's keynote speakers. They include:

- Professor Ed Crawley, SDM cofounder, professor of aeronautics and astronautics and engineering systems, and Ford professor of engineering (Monday, October 24)
- Julian Goldman, MD, medical director of biomedical engineering at Partners HealthCare Inc.; founding director and principal investigator, Medical Device Interoperability Program at the Center for Integration of Medicine and Technology; attending anesthesiologist, Massachusetts General Hospital/Harvard Medical School (Tuesday, October 25)

The event will provide significant opportunities to ask questions of all of our speakers, as well as to network with other systems thinkers attending the conference. An evening reception will be held October 24 for all who hold full conference admission.

Additional speakers and registration information will be posted in early summer at sdm.mit.edu.

For information on corporate sponsorship, contact SDM Industry Codirector Joan S. Rubin at jsrubin@mit.edu or 617.253.2081.

Webinar series offers insight into applied systems thinking

By Lois Slavin, SDM communications director

The MIT SDM Systems Thinking Webinar Series provides information on how systems thinking can be applied across technical, managerial, and socio-political domains to address complex problems in a wide range of industries, as well as nonprofit and governmental organizations.

Each webinar features research conducted by SDM faculty, alumni, students, and industry partners that illustrates how to employ systems thinking in real-world situations.

Held twice monthly from noon to 1 pm, the series is open to all at no charge. Recorded webinars are also available on demand at sdm.mit.edu. Click on Voices in the upper nav bar.

Coming up on March 14, 2011, MIT Senior Lecturer Steve Spear will present "Leadership, Innovation, and Operational Excellence: How Market Leaders Beat the Competition." Spear, who is also a senior fellow at the Institute for Healthcare Improvement, is an internationally recognized expert in leadership, innovation, and operational excellence.

On March 28, 2011, John Kluza, business development manager for emerging applications at Satcon Technology, will present "Status of Grid-scale Energy Storage and Strategies for Accelerating Cost-effective Deployment." Kluza is an SDM alumnus and a former energy storage and smart grid analyst.

SDM alum Akshat Mathur will present a webinar April 25, 2011, on "The Evolution of Business Ecosystems: Interspecies Competition in the Steel Industry." Mathur is an accomplished operations and supply chain management professional who has worked in various operations management, strategy, and planning roles in the steel industry in India.

For further information, please contact SDM Industry Codirector Joan S. Rubin at jsrubin@mit.edu or 617.253.2081.

Joan Rubin appointed SDM industry codirector

By Lois Slavin, SDM communications director

Joan Rubin

MIT's System Design and Management Program (SDM) has announced that Joan Rubin has joined SDM as industry codirector. Rubin comes to MIT from Covidien, a leading manufacturer of medical devices and supplies, diagnostic imaging agents, and pharmaceuticals, where she served as vice president of business development.

Rubin brings to SDM 17 years of business development, marketing, market development, and strategic planning experience in the medical device field. She joined Aspect Medical Systems in its startup phase several years prior to its November 2009 acquisition by Covidien. At Aspect, her roles included vice president of business development, senior director of global partnerships, director of global upstream marketing, and manager/director of market development.

In building Aspect from a startup to a profitable public company, Rubin was responsible for cultivating, nurturing, and sustaining partnerships with many large, global companies. "I am looking forward to identifying ways that we can further develop connections between industry and the SDM program," she commented. Previously she worked as manager of surgical marketing at Haemonetics Corp.

Professor Warren Seering, SDM codirector from MIT School of Engineering noted, "Joan's deep knowledge of the medical and pharmaceutical industries, strong relationship management skills, and senior-level contacts are enormous attributes. We very much look forward to working with her and to continuing to evolve SDM's research, academic, and conference offerings to meet the needs of a wide range of industries."

"Joan brings to SDM a unique blend of current connections with a broad range of global multi-billion-dollar companies as well as small, emerging technology companies that will influence industry in the years to come," said Steven Eppinger, SDM codirector from MIT Sloan School of Management. "We welcome the opportunity to build relationships that can lead to their participation in SDM."

Rubin is a graduate of MIT's Leaders for Global Operations Program, where she earned an SM in management from MIT Sloan and an SM in mechanical engineering. She holds an ScB in mechanical engineering from Brown University. Rubin and her husband, Dan, have two children.

SDM students launch MIT chapter of INCOSE

By Donny Holaschutz, SDM '10

Donny Holaschutz, SDM '10

Students in the MIT System Design and Management Program (SDM) this fall launched the MIT Student Division of the International Council on Systems Engineering (INCOSE).

INCOSE is a not-for-profit organization dedicated to advancing the field of systems engineering for the benefit of humanity. The student chapter was founded to create opportunities for students at MIT to become involved in INCOSE, and to provide INCOSE members with access to the innovative and entrepreneurial spirit of MIT.

The chapter was founded by several members of SDM's 2010 cohort—Karl Critz, Matt Harper, Donny Holaschutz, Rafael Marañón-Abreu, and Arjun Shrinath—along with Troy Peterson, SDM Certificate '09. Associate Professor Olivier de Weck, the associate director of the MIT Engineering Systems Division (within which SDM resides), is the group's faculty advisor.

The student chapter at MIT enriches a long-standing partnership between SDM and INCOSE that dates back to the organization's founding in 1990. Several MIT research staff and professors have held leadership roles in the INCOSE national organization over the years, including Pat Hale, director of the SDM Fellows Program, who is a past president of the organization.

The MIT Student Division of INCOSE is dedicated not only to advancing systems engineering practice through its activities, but also to developing tomorrow's systems thinkers and leaders. Activities held this fall included presentations by de Weck and Neil Snyder, the executive director of systems engineering and program integration at the National Renewable Energy Lab.

Massachusetts Institute of Technology

SDM calendar spring–fall 2011

If you or your colleagues are interested in attending any of the events listed, please contact SDM Industry Codirector Joan S. Rubin at jsrubin@mit.edu or 617.253.2081.

March 14, 2011
MIT SDM Systems Thinking Webinar Series
Leadership, Innovation, and Operational Excellence: How Market Leaders Beat the Competition

Steve Spear, MIT senior lecturer and senior fellow of the Institute for Healthcare Improvement

Location: sdm.mit.edu

Time: Noon-1 pm

Details: http://sdm.mit.edu/voices/webinars.html

March 28, 2011
MIT SDM Systems Thinking Webinar Series
Status of Grid-Scale Energy Storage and Strategies for Accelerating Cost-Effective Deployment

John Kluza, SDM '08, business development manager for emerging applications at Satcon Technology

Location: sdm.mit.edu

Time: Noon-1 pm

April 12, 2011
SDM Information Evening
Location: Boston Marriott Burlington

Time: 6-9 pm

April 25, 2011
MIT SDM Systems Thinking Webinar Series
The Evolution of Business Ecosystems: Interspecies Competition in the Steel Industry

Akshat Mathur, SDM '08

Location: sdm.mit.edu

Time: Noon-1 pm

June 14, 2011
SDM Information Evening
Location: Marriott Quincy

Time: 6-9 pm

June 20-23, 2011
Annual INCOSE Symposium
ESD/SDM exhibit

Location: Denver, CO

October 23, 2011
SDM Alumni-Student Networking Mixer
Details: TBA

October 24-25, 2011
MIT SDM Conference on Systems Thinking for Contemporary Challenges
Keynote speakers: Ed Crawley, SDM cofounder and MIT professor of aeronautics and astronautics and engineering systems, as well as Ford professor of engineering; Julian Goldman, MD, medical director for biomedical engineering at Partners HealthCare Inc.

Details: See page 18

October 26, 2011
SDM Industry Partners Meeting
Details: TBA

October 26, 2011
SDM Information Evening
Location: MIT Faculty Club

Time: 6-9 pm

Event information includes all details available at press time. For more current event information, go to sdm.mit.edu and esd.mit.edu.