



**Michael Gray has nine  
months to overhaul systems  
and software development.**

In 15 minutes you'll know how he does it.

# What if...

*... you had to meet process quality standards in just nine months or lose contracts with your biggest customer?*

*... you had to get the latest product upgrade to market faster than your competitors or risk losing market share?*

*... you had to adopt distributed development or lose your competitive edge?*

## And what if...



your development team told you, “We can’t do it. We don’t have the right infrastructure in place or the process to make this happen.”

What would you do?

At Telelogic, we know what’s going on in boardrooms and conference rooms around the world as companies endeavor to improve their development process and battle to be first to market with higher quality products. More importantly, we know how to help companies attain these objectives using our integrated suite of market-leading application development lifecycle tools.

To illustrate why our systems and software development tools are preferred by leading companies in vertical markets like telecommunications, aerospace/defense, automotive, financial services and medical device manufacturing, we’ve created a fictional story based on real customer experiences.

Don’t be surprised if it sounds familiar.

Does it have a happy ending? Read on to find out...

## Process and Productivity Improvement Meeting

In attendance:

Michael Gray  
CIO



Aaron Katz  
Systems Engineering Lead  
for Requirements Analysis



Kala Patel  
Lead System  
Architect



Victor Chang  
Configuration  
Management  
Director



Gary Ofstroski  
Project Team Leader



Julie Sanchez  
Software  
Development  
Manager



Jim Nelson  
CFO



The group seated around the table in the north conference room falls quiet as a last-minute arrival closes the door and takes a chair. Michael Gray, the CIO of a leading global manufacturer, looks up from the notes he's made for this morning's meeting.

“Good morning. I hope you had a nice weekend. Everyone's here, so let's get started. Now that the acquisitions of Movarian Systems and Storenica are finalized and we've got our team in place, I want to talk about where we are and where we need to get to this year.

“I'll get right to the point. We need to meet some specific process quality standards to keep our approved vendor status with some of our biggest customers. **They're going to audit our development process and it's imperative we start using a more mature process than what we have now.** The executive management team has mandated that we've got to get it rolled out in nine months. So process improvement's going to be our mantra from now on. Also, we're going to have to take the plunge into distributed development to take advantage of the development resources we've acquired in India and Lithuania. Lastly, we've got to reduce cycle time somehow. Zytron Industries is killing us. They just released the BX73 and it's only been six months since the 72 came out.”

Michael scans the faces of the team he's gathered in the conference room. **The company's future success depends on their expertise and experience.** Do they have what it takes? What can they bring to the table?

*Julie Sanchez, the company's software development manager, shakes her head. “No way, Michael. I mean, meeting a formal process quality standard in nine months? It's just not going to happen. And don't even think about getting Merlin out the door in six months. We'll be lucky to make the 12-month deadline.”*

“Why? What's stopping us?”

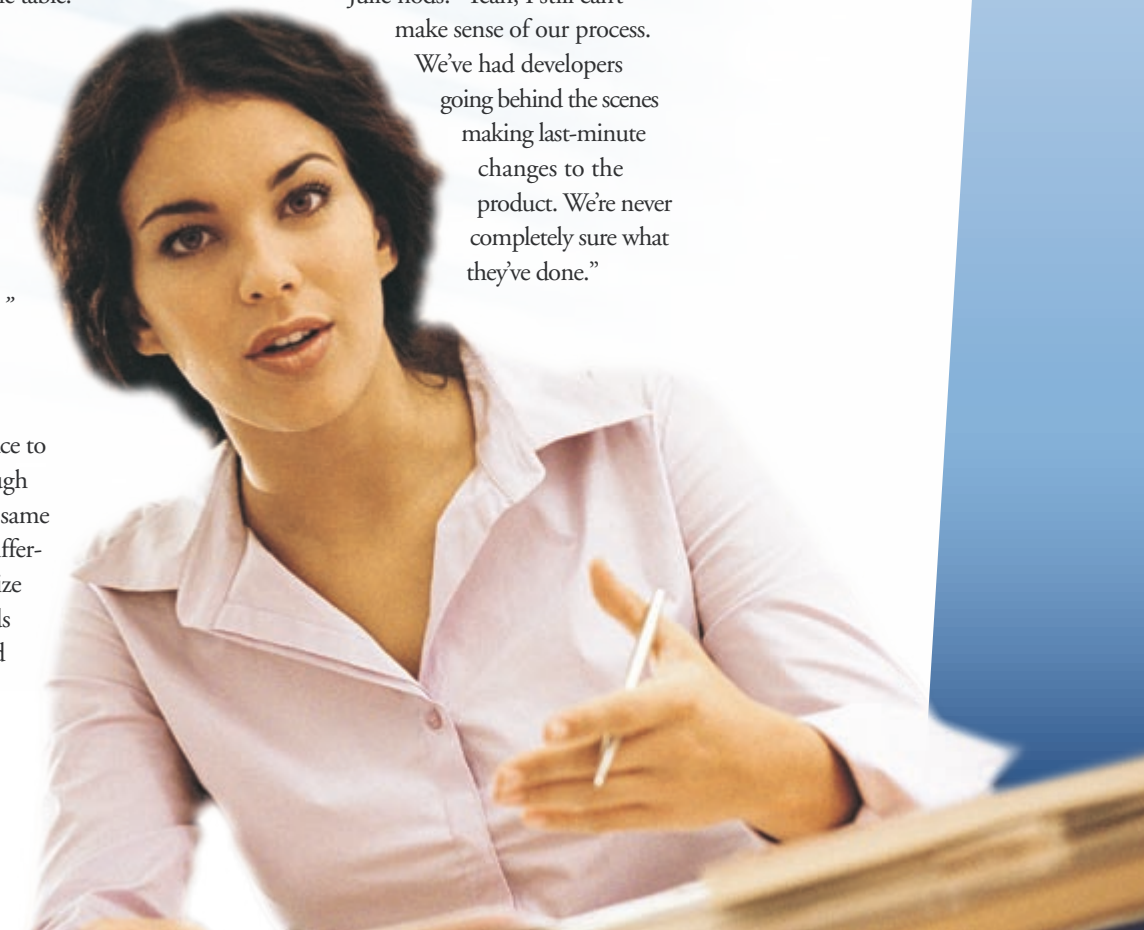
“We don't have the right infrastructure in place to support improved processes. It's a big enough challenge to get all the groups to follow the same process, not to mention there are a dozen different tools being used. We'd have to standardize on a process and automate it to see the kinds of improvements you're talking about. And that means standardizing on tools as well.”

Gary Ofstroski looks up from his notebook. “Julie's right, Michael. After these acquisitions, we've got too many tools that just aren't integrated. Our resources are stretched too thin. We're also weak on some capabilities. Distributed development is beyond us at this point. We've got to decide which tools we're going to standardize on, but how? Should we just throw everything out and start from scratch?”

“I don't know if starting over is the solution or not,” says Victor Chang. “But I do know that change and configuration management are virtually non-existent here and we're paying the price. On the Sierra project we were five weeks from the delivery date and somehow we found a bug we were sure we'd fixed ages ago. **We had developers pointing fingers at each other. It takes me hours every day just to keep track of who changed what!** We had a hard time reproducing last week's build and we are still not sure that what we produced is exactly the same as what we sent our beta customers.”

Julie nods. “Yeah, I still can't make sense of our process. We've had developers going behind the scenes making last-minute changes to the product. We're never completely sure what they've done.”

“A distinguishing characteristic of Telelogic's tools is that they are well suited for large-scale development projects for highly complex, advanced systems and software, often with geographically dispersed development teams.”  
– “Application Development Vendor Analysis,”  
Ovum, Clive Burrows, July 2003.



## “IT KILLS US WHEN WE OVERLOOK REQUIREMENTS...”



“That’s a really big problem,” Michael says, “But I’m actually more worried about requirements management – or the lack thereof – and its impact on our ability to manage our biggest contracts, as well as our OEM subcontractor relationships. **Every morning I see developers working away and I need to be confident that what they’re working on relates to the most current customer specs.** It kills us when we overlook requirements and then have to bolt on some solution weeks before delivery or miss milestones and push the delivery date back. We have to be able to prove to our customers that all requirements were worked on, tested and are fully traceable back to the original contract. If we can’t prove that we’ve satisfied their requirements, we don’t get paid, we lose business and then... well, you know the rest.”

### The group falls silent.

Losing some of these lucrative contracts is unthinkable. Their jobs are on the line.



Aaron Katz raises his hand. “This kind of reminds me of the requirements challenges we were facing where I worked before. **We had five different development sites around the country and one in India all needing to access the same requirements data.** On this one project we had over 100,000 requirements and we were using Excel and Word. It was a complete nightmare, especially when the requirements changed. Then I found that one of our subcontractors was using a requirements management product called DOORS.

“It really helped because we could move all our requirements from Excel and Word or whatever into DOORS. After that everyone could access and work from the same set of requirements.”

Julie shrugs. “But you can do that with Excel or Word.”

“Not really. With DOORS you have complete traceability from the high-level user requirements that you’re contracted to meet right down to the detailed specs that define how you’re going to build the system. So it’s real easy to verify and validate – you can see all related requirements side by side. You can also get a handle on the impact of any proposed changes to the user requirements on your specs. DOORS also records every requirements change and subsequent actions for the entire project. That’s the beauty of requirements-driven development.”

*Jim Nelson leans forward. “Building this kind of infrastructure in six locations sounds like a huge investment.”*

“**Actually, a small investment compared to the cost overruns and quality issues we were dealing with – and it paid off fast.** We saw significant ROI from DOORS right from the very first project.”

**Jim nods. “We could use some of that around here.”**

“Another thing about DOORS is that each requirement is a separate object that has not only the text of the requirement, but additional information called attributes like ‘priority’ or ‘cost’ that you can define. If you want to add additional information to the requirement, maybe something like ‘importance’ or ‘completed’, you just create an attribute and assign values to it.

“**DOORS also allows you to control who sees what –** that means all the different stakeholders in the project can look at the information from different perspectives. You just click on the view you want to see from a pull-down menu and DOORS displays all the requirements and attributes that you have permission to view and lets you link them together. It’s really great.

“And going back to Word and Excel, **DOORS can import and export requirements** to and from both of them as well as lots of other formats.”

“You know,” Michael says, “having a requirements-driven methodology that ties together our different sites, teams and outside contractors would be a huge step forward.

“Telelogic DOORS is the market share leader for requirements management products and has the most complete life-cycle integration.”

- “Mastering the Requirements of Requirements Management: Application Delivery Strategies, Integration & Development Strategies,” META Group, Thomas Murphy, April 2003.

“Aaron, how did you create paper reports? You know that’s what our customers want. **Right now we spend too much time cutting and pasting information from a lot of different tools in different formats.**”

“We used DocExpress,” Aaron says. “It’s from the same company that makes DOORS. “DocExpress is basically a report generator – you build documents from data in the tools rather than having to push info into the tools to generate the documents. It saved us a ton of time.”



“...CHANGE AND CONFIGURATION MANAGEMENT ARE PRACTICALLY NON-EXISTENT AROUND HERE.”

8:56am

Michael nods. “Good input, Aaron, thanks. Anyone else?”

Gary straightens up in his chair. “You know, Victor said earlier that change and configuration management are practically non-existent around here. It was like that at the telecom company I used to work for. They were getting into distributed development for the first time.

**After a couple of false starts, they brought in this CM suite called SYNERGY and deployed it throughout the enterprise. It was really scalable and flexible.** They used it to manage small rapid development projects as well as the complex global development ones.

“And Julie, you’ll be interested in this. SYNERGY/CM is much easier to use than any other CM tool out there. Those users of yours who don’t use CM tools ‘because they’re too complex and they take too much time’ will be pleasantly surprised. You can use SYNERGY to track all of your developers’ work assignments, or as SYNERGY calls them, tasks. **Each of your developers will have an individual to-do list where he or she can see and prioritize task assignments.** When they’re ready to start working on an assignment they simply select it, and once they’ve finished, they mark it as completed. Meanwhile, SYNERGY will take care of automatically linking their edits to the task, documenting the change and delivering the files they worked on. Your developers will quickly see how much time they’ll save, believe me.

“And Julie, there’s more good news for you: **SYNERGY’s task-based approach means you can see the status of all work assignments in real time,** for instance all completed work for your project. You can then just pull these changes in to complete the latest build and in a single click, view all the features added and bugs corrected. And if a destabilizing change is introduced, you can correct the problem by simply removing the troublesome task from the configuration. It’s much cleaner than having to worry about individual files!”

“That kind of visibility and control sounds just like what I need,” Julie says.

*Michael taps his finger on the tabletop thoughtfully. “As managers, we’ve got to get a handle on what’s going on in the development stream. Right now it’s all e-mails and phone calls. Can SYNERGY really give us that level of visibility and support globally distributed teams?”*

“Well,” Victor says, “we had over 1,000 developers all over the globe using SYNERGY and there was never a scalability issue. Every morning the developers would come in, get all the latest changes in a single operation and go off and do their work. So they were always on top of the latest changes, and as managers, we were sure what they were doing was within the scope of the project. Also with SYNERGY, you can associate the original requirements or change requests to the actual work assignments, as well as link these tasks to the relevant objects. That gives you major visibility and traceability through just one interface.”



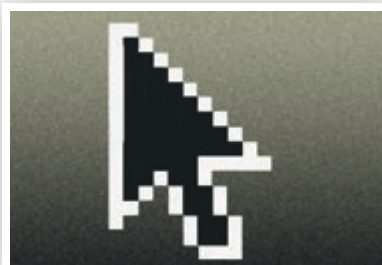
“So what about parallel development support?” Michael says. “That’s high on my list this year.”

“SYNERGY made implementing parallel development a lot easier because it automatically notified developers of parallel versions on check-out and check-in. It also guided them to merge those versions when necessary.”

“[SYNERGY/CM] is the market leader and demonstrated a clear differentiating comparative advantage.”  
– “Application Change Management Software, 2002,”  
Yphise, Laurent Mynard, January 2003.

9:14am

“...WE COULD TRACK EXACTLY WHAT CHANGES WERE INCLUDED IN ANY GIVEN BUILD, RELEASE OR CONFIGURATION.”



Gary raises an eyebrow. “So how did you deal with making sure that bug fixes stayed fixed in later versions?”

“That’s an easy one – you simply apply the bug fix task to the parallel release stream.”

“You’re getting too technical here!” laughs Michael. “Pretend I don’t understand...”

“Sorry... I mean that I just add that bug fix to the list of changes to pull in for the project and SYNERGY takes care of the rest.”

“We can’t do that now,” Julie says, “because **there’s nothing managing the relationship of the files with the reason for the change.**”

“How come?” Michael says.

“Our tools don’t support it. We’d have to write so many scripts to link everything together and we just don’t have the time or resources.”

“Victor, does SYNERGY have this linking capability?” asks Michael.

*“Yes. We used to have bugs reappearing from release to release, but once we implemented SYNERGY, we were able to eliminate the problem completely and we could track exactly what changes were included in any given build, release or configuration.”*

**“What about integration with our existing tools, like Microsoft Project and Mercury TestDirector?” Julie says.**

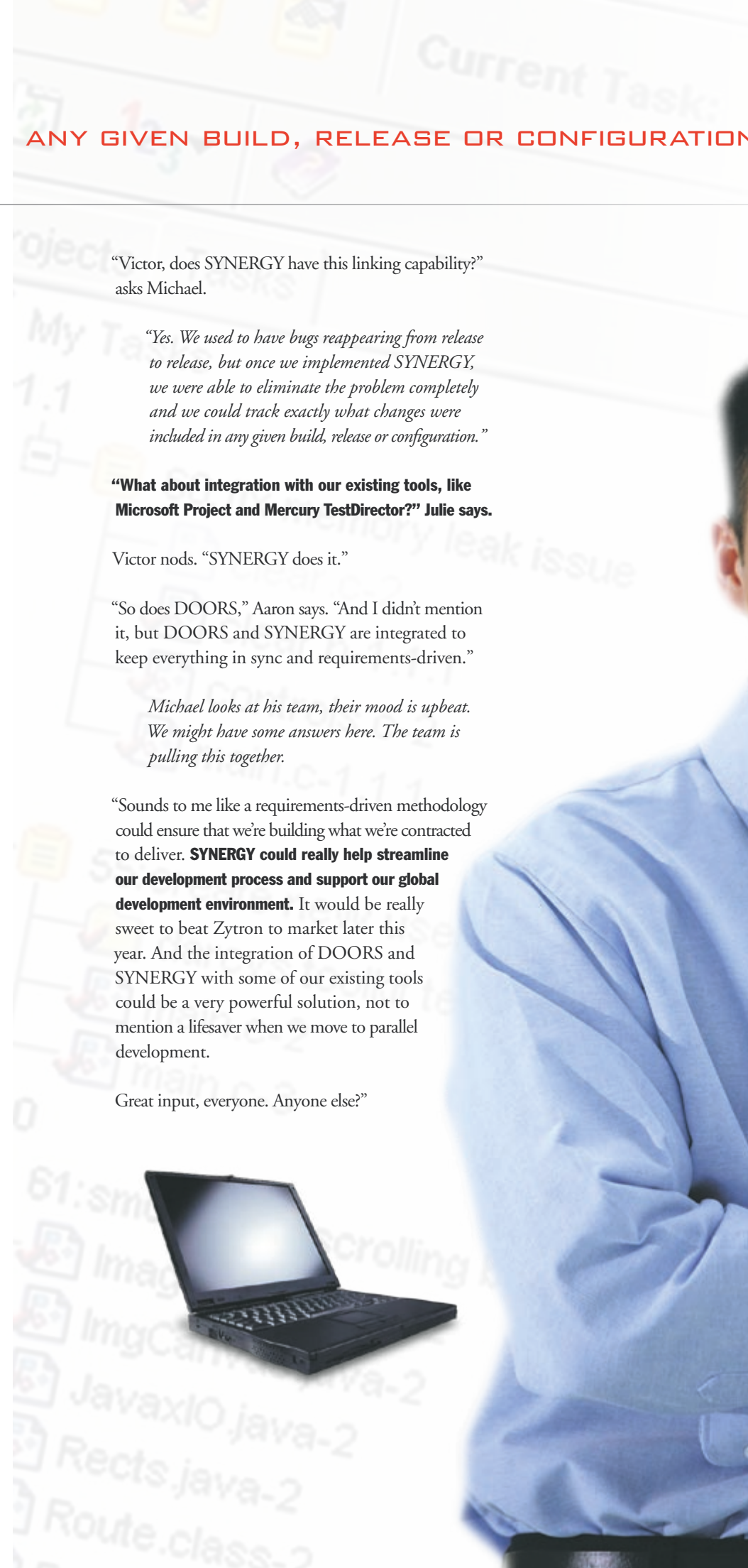
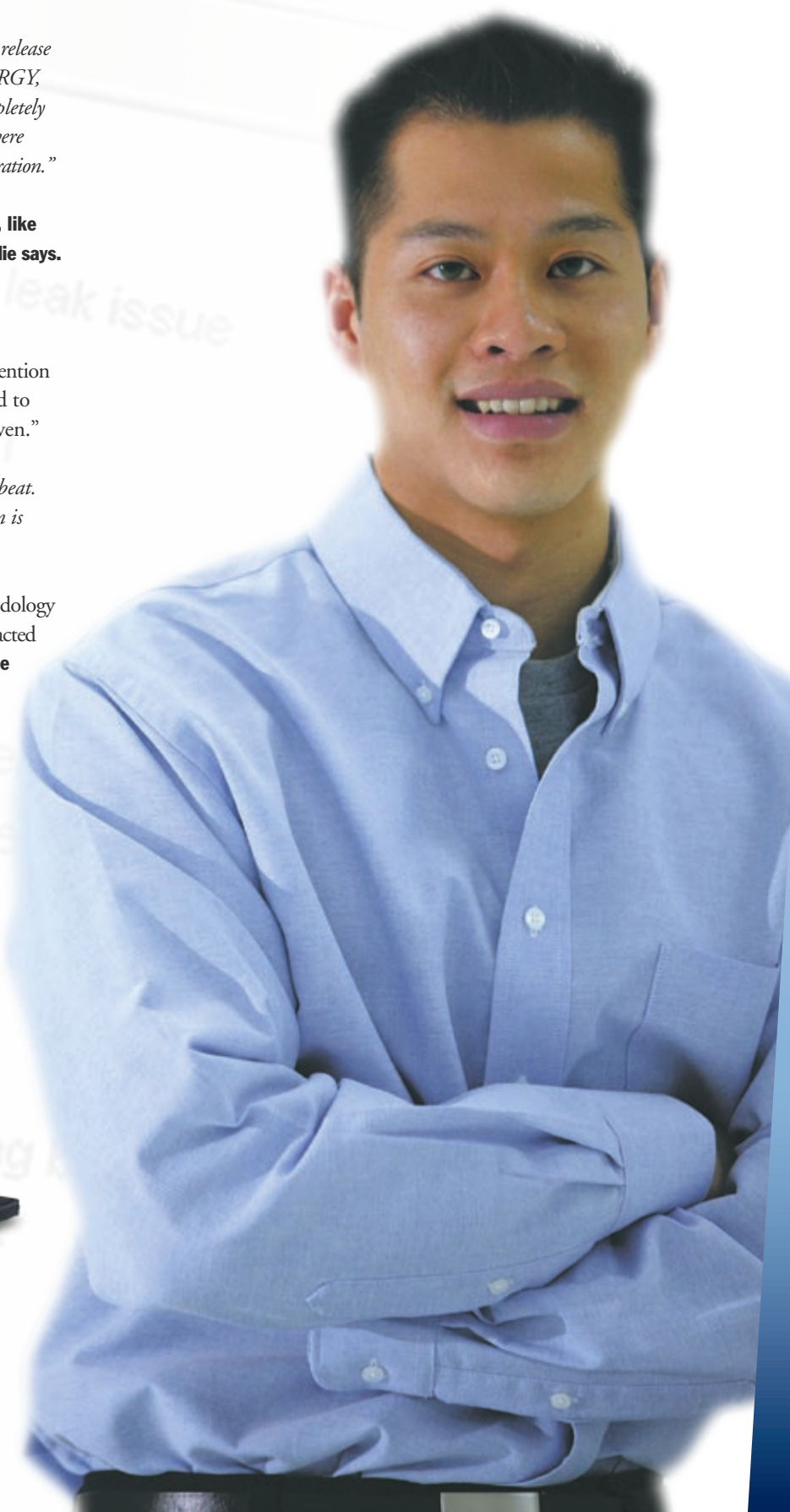
Victor nods. “SYNERGY does it.”

“So does DOORS,” Aaron says. “And I didn’t mention it, but DOORS and SYNERGY are integrated to keep everything in sync and requirements-driven.”

*Michael looks at his team, their mood is upbeat. We might have some answers here. The team is pulling this together.*

“Sounds to me like a requirements-driven methodology could ensure that we’re building what we’re contracted to deliver. **SYNERGY could really help streamline our development process and support our global development environment.** It would be really sweet to beat Zytron to market later this year. And the integration of DOORS and SYNERGY with some of our existing tools could be a very powerful solution, not to mention a lifesaver when we move to parallel development.

Great input, everyone. Anyone else?”



“...YOU CAN AUTOMATICALLY GENERATE CODE IN C, C++ OR JAVA FROM THE MODEL.”



“Yes,” says Kala Patel. “I’d like to discuss visual modeling. The tool you have now doesn’t support UML 2.0 and I need that if you expect me to do any kind of architectural diagramming.”

“UML what?” asks Jim.

### Aaron jumps in.

“UML stands for Unified Modeling Language and version 2.0 is the latest standard. **It’s great because it would allow our systems engineers to ‘speak’ the same visual language as our design teams.**”

“The requirements people would love to visualize or model requirements using UML,” Aaron says.



“Well, Aaron mentioned DOORS earlier,” Kala says. “Where I used to work we used DOORS/Analyst. Our systems engineers would analyze requirements and do some basic system and requirements modeling all within DOORS/Analyst. I could take those models and further refine them with architectural and component diagrams and even go into some pretty heavy-duty simulation and validation capabilities.”

### Aaron smiles. “Wow! That’s light years ahead of what we have now.”

“Yeah, Telelogic has role-based tools for DOORS as well as TAU Generation2, a suite of systems and software tools for development and testing,” says Kala. “We’d start off with DOORS/Analyst, refine with TAU/Architect and then complete the development in TAU/Developer. You can use executable models to do dynamic and behavioral modeling without having to write any code. With real-time and embedded applications, once you validate the behavior you can automatically generate code in C, C++ or Java from the model. So we kept hand-coding to a minimum and saved maybe six months right off the bat.”



Julie taps some notes into her laptop. “Sounds like DOORS/Analyst, TAU/Architect and TAU/Developer could help better integrate system engineering and software development. They’re two silos right now. Breaking them down would definitely improve our process and help us pass a formal assessment by that nine month deadline. **And if we used TAU/Developer to automatically generate most of our code, we could cut our cycle time by 50 percent** and at least match Zytron.”

*“Going back to what we were talking about earlier,” Michael says. “How can we be sure our developers are working on the current customer requirements?”*

Kala interlaces her fingers and smiles. “You can integrate DOORS with TAU. In TAU there’s a tab you can click on and your requirements are right there. Then you drag and drop requirements to model elements to establish traceability and synchronize the linking you’ve made back to DOORS. **Everyone gets to work with their own interfaces, but you still have communication across the project and between the tools.** At my old company, our designers and developers always had visibility of the requirements and could easily link them with UML design elements to show where and how the requirements were satisfied by the design. If anyone’s interested, we went one step further and also used TAU/Tester. Actually, when I think about it, TAU was pretty much everywhere.”



"Yeah, I know what you mean," Victor says. "Where I worked we had all the TAU tools, including Logiscope, which was a great quality assurance tool. It let us analyze and document legacy code so we could have a better idea of what could be re-used. In fact, we used all of the tools we've been talking about: DOORS, TAU and SYNERGY. Because the tools integrate, you can go all the way from requirements to code with UML 2.0 using DOORS/Analyst and TAU. Then you can manage change requests and configurations with SYNERGY throughout the development lifecycle. It actually delivers that requirements-driven development that organizations are always talking about, but have never been able to do."

**"From requirements to code using one standard notation – that's a dream come true for managing the development lifecycle,"** Michael says as he writes on his note pad.

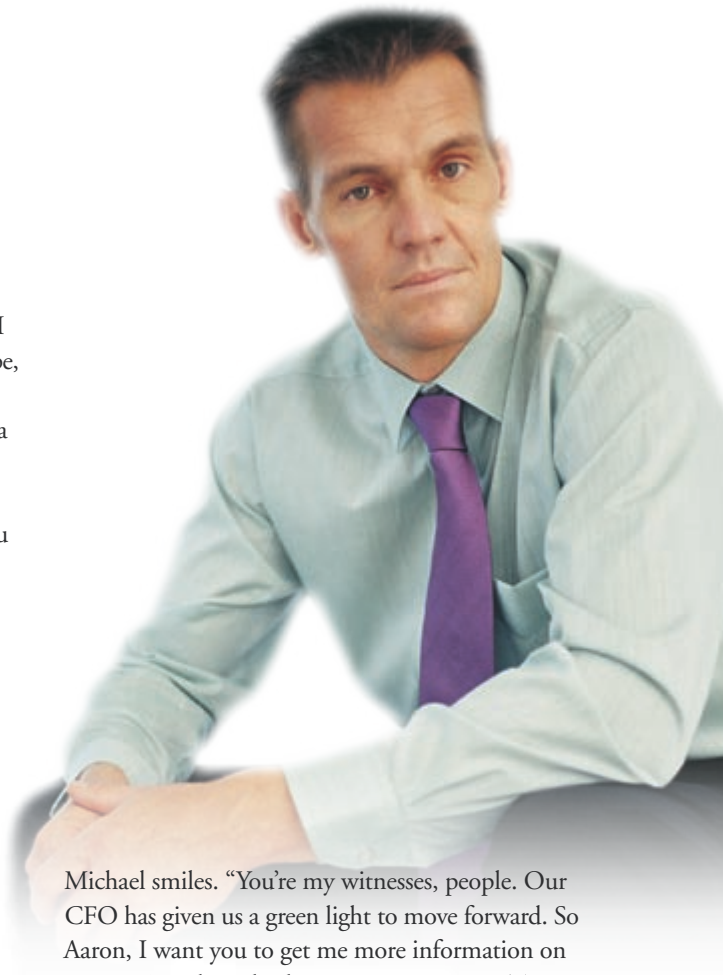
*"And you know, Michael," adds Aaron, "from what I have seen of DOORS, it would be really useful for managing some of our higher level company requirements. I mean, corporate goals and objectives are just another kind of requirement, aren't they? How about applying the concept of traceability to program management? I'd sure like to understand a little better the relationships between our projects and exactly how they contribute to the overall program. And we need to manage the contracts with some of our suppliers a little better."*

"Among its strengths is its support for teams of distributed modelers and developers. TAU/Developer and TAU/Architect were some of the first tools to contain features that have now become part of UML 2.0."

– "The OOA&D Market: 2004 Vendor Update Details," Gartner, Michael Blechar, August 2003.

Michael looks thoughtfully at the ceiling. A more disciplined approach to defining company strategy would be a great idea, especially when things are changing so rapidly with the new acquisitions.

"Well, Michael," Jim says, "sounds like you and your team have some homework to do. If these tools really do what your team says, get me some numbers and I'll move fast and speak to the CEO."



Michael smiles. "You're my witnesses, people. Our CFO has given us a green light to move forward. So Aaron, I want you to get me more information on DOORS. Kala and Julie, I want a report on TAU from you. Victor, please put together what you can about SYNERGY. Gary, I'd like you to get back to me with some quantifiable benefits you think we could gain here."

Aaron frowns. "But who makes all these tools?"

**"They're all from the same company – Telelogic," Kala says, looking at her laptop. "I've got their Web site bookmarked right here – [www.telelogic.com](http://www.telelogic.com)."**

Michael glances at his watch. This has to have been the most productive meeting of the year.

"I've got another meeting, so let's wrap this up," he says. "Great work this morning, everyone. I think we may be on the way to getting our process on track and improving productivity. Have those reports to me by Friday morning and we'll plan our next steps at next Monday's team meeting."

**"Thanks!"**

Find out how Telelogic can help you develop better systems and software at less cost. Visit us at [www.telelogic.com](http://www.telelogic.com) for more information on our products and lifecycle solutions or download our interactive CD-ROM at [www.telelogic.com/guide](http://www.telelogic.com/guide).

gpsnav-dar

Name

gpsnav

gpsnav

gpsnav

chart

.class

.project

123.eps

5perc\_s

a1

AboutDialog

AboutDialog

all.wpt

Ann.class

Ann.java

avvp.class

avvp.java

browser





**World Headquarters**

P.O. Box 4128 • SE-203 12 Malmö, Sweden • Phone: +46 40 650 00 00 • Fax: +46 40 650 65 55

**American Headquarters**

9401 Jeronimo Road • Irvine, California 92618 USA • Phone: +1 949 830 8022 • Fax: +1 949 830 8023

© 2004 Telelogic AB. Telelogic, Telelogic DOORS, Telelogic TAU and Telelogic DocExpress are the registered trademarks of Telelogic.  
Telelogic DOORS/Analyst, Telelogic TAU/Architect, Telelogic TAU/Developer, Telelogic TAU/Tester, Telelogic TAU/Logiscope and Telelogic SYNERGY are trademarks of Telelogic.  
UML and Unified Modeling Language are trademarks of the Object Management Group.  
All other trademarks are the properties of their respective holders.