XP2013, Per Branger, 2013-06-04

Agile Practices and Training in ABB

A global leader in power and automation technologies
Leading market positions in main businesses

- 145,000 employees in about 100 countries
- $39 billion in revenue (2012)
- Formed in 1988 merger of Swiss and Swedish engineering companies
- Predecessors founded in 1883 and 1891
- Publicly owned company with head office in Switzerland

Shaping the world we know today through innovation
Pioneering technology since 1883

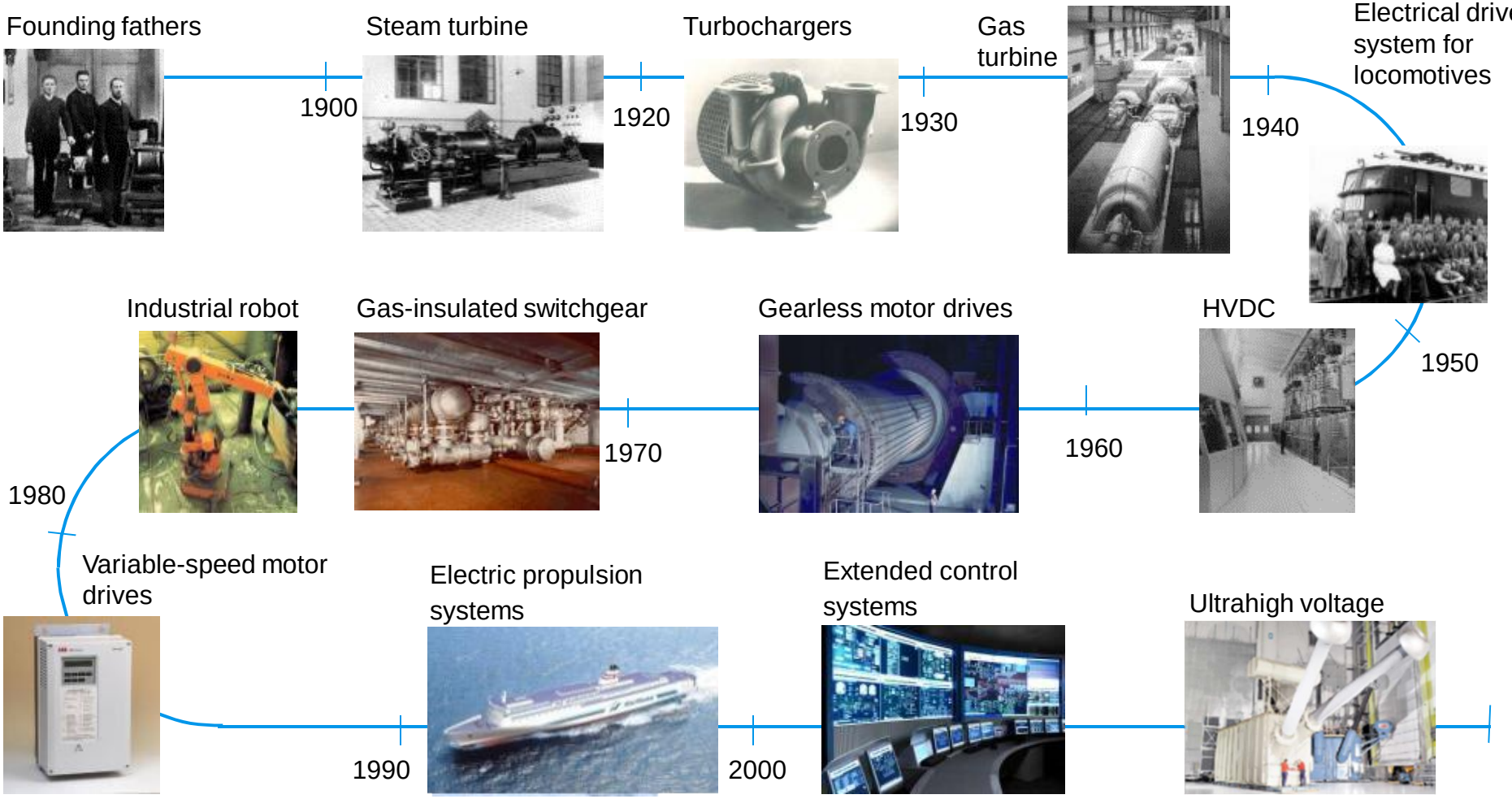


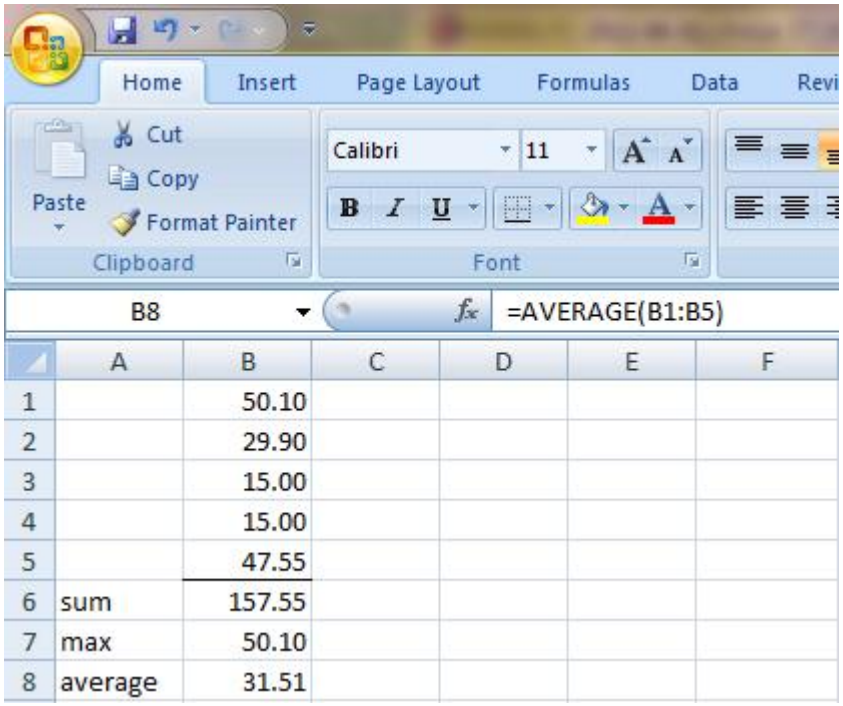
ABB technology
Software ascendance

"Can ABB become recognized as a software company?"

- 1891: BBC starts selling boxes
- 1980s: first DCS
- 2000s
  - Industrial IT, 800xA, SKYVA (till 2003)
  - Network Manager
- 2010s
  - Ventyx, Mincom
  - 2500 software developers
- 2013: software community of 10'000 people

Computer science expertise is essential for ABB
(service, security, architecture, integration, acquisitions, …)

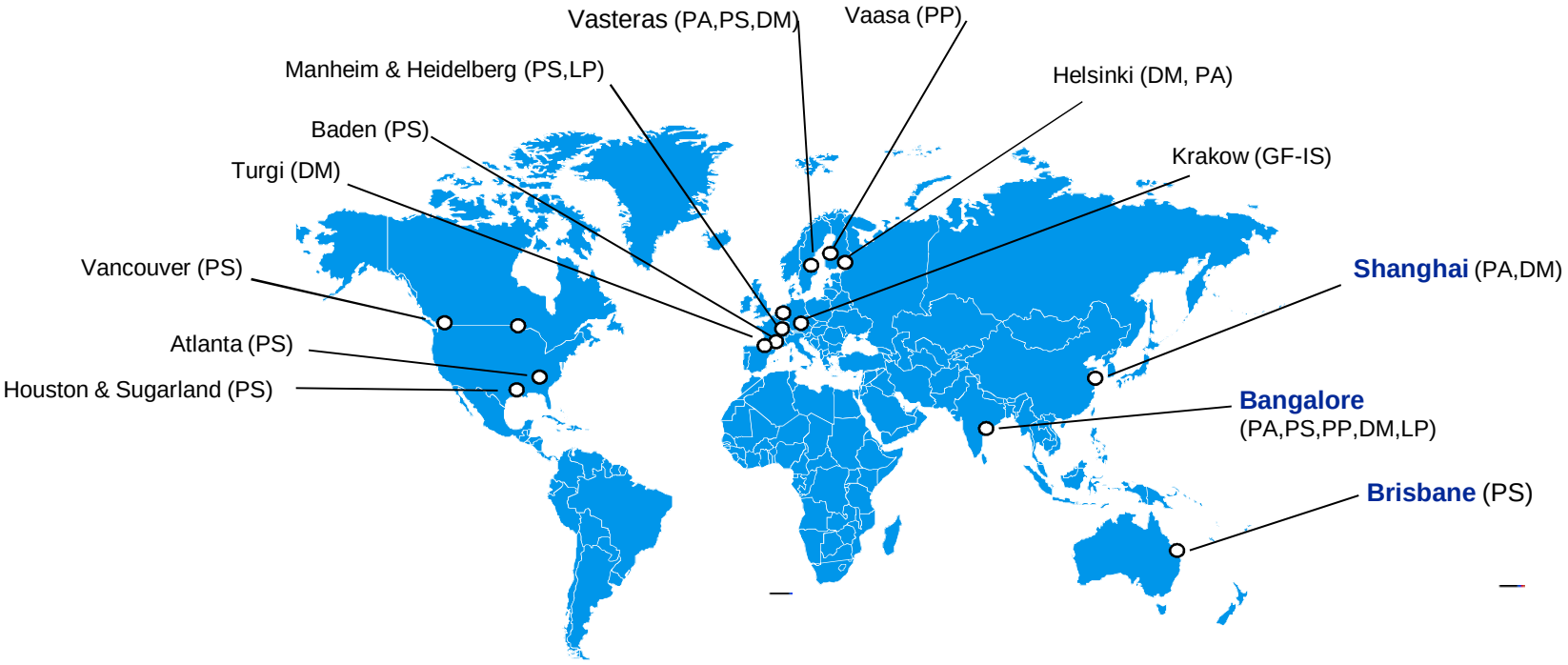
Complexity: Excel vs. ABB controller

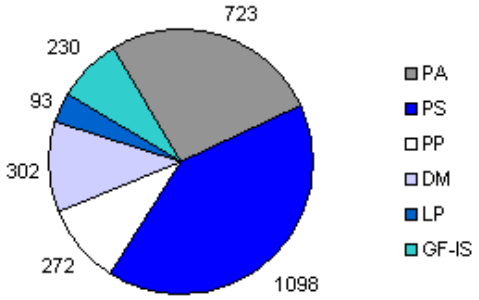


	A	B	C	D	E	F
1		50.10				
2		29.90				
3		15.00				
4		15.00				
5		47.55				
6	sum	157.55				
7	max	50.10				
8	average	31.51				

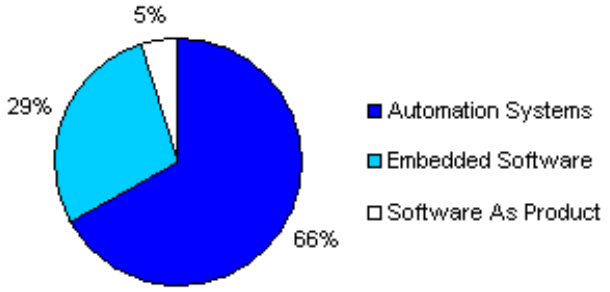
30–50 MLOC

Distributed development



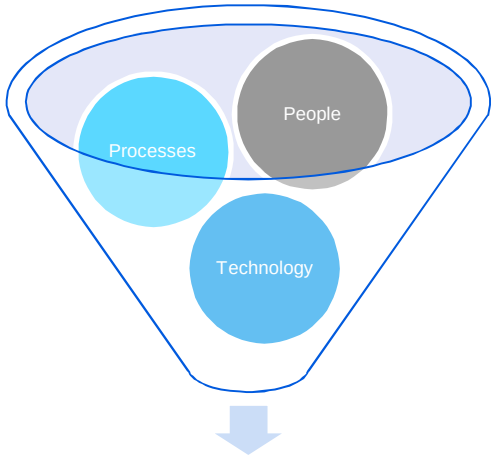


Software Engineers Per Division



Distribution Of Revenue

Software Development Improvement Program
Transforming Software Development in ABB



- SDIP was launched in 2008 as an ABB Group initiative and chartered to **transform the way ABB develops software**
- For the benefit of ABB's overall business objectives, SDIP aims to bring our software R&D above and beyond industry average to achieve speed, quality and predictability in ABB's software product development.
- Core Team consists of 3 people at ABB Group and 5 division representatives
- Local improvement initiatives in Divisions and Business Units

Big Three Transformations

Transforming the way we manage our portfolio:

Continuous Portfolio Management

Transforming the way we release products:

Continuous Release Management

Transforming the way we develop products:

Continuous Development

Practices 2013
Basic and Advanced

Basic (Mandatory) Practices
Introduction of Configuration
Introduction of Requirements
Introduction of Architecture
Introduction of SDIP Tools
Static Code Analysis
Code Review
Coding Standards
Unit Testing with Automation
Project and Product Metrics
Software Estimation
Nightly Builds
R&D Self-Assessment
Define and Control Interfaces

Advanced (Encouraged) Practices
PPQA Audits
Portfolio Planning with Themes
Product Visions and Roadmaps
Fixed Cycle Release Planning
Features, Scenarios and User Stories
Scrum
Functional Test Automation
Design with Iterative Prototyping
Source Code Refactoring
Continuous Integration

Practices

Site Actions | Browse | Page | Sascha Stoeter

SDIP Portal ▸ Practices | This Site: SDIP Portal | I Like It | Tags & Notes

SDIP Portal | PA Portal | DM Portal | LP Portal | Cyber Security

Topics

- Practices
  - Training
  - Tools & Services
  - People
- **Sharing**
  - Knowledge Bits
  - Productive Bits
  - SDIP Blog
  - Code Bits (soon)
  - Resources
- **Examples**
  - RAISE
  - Success Stories
- **About**

Practices

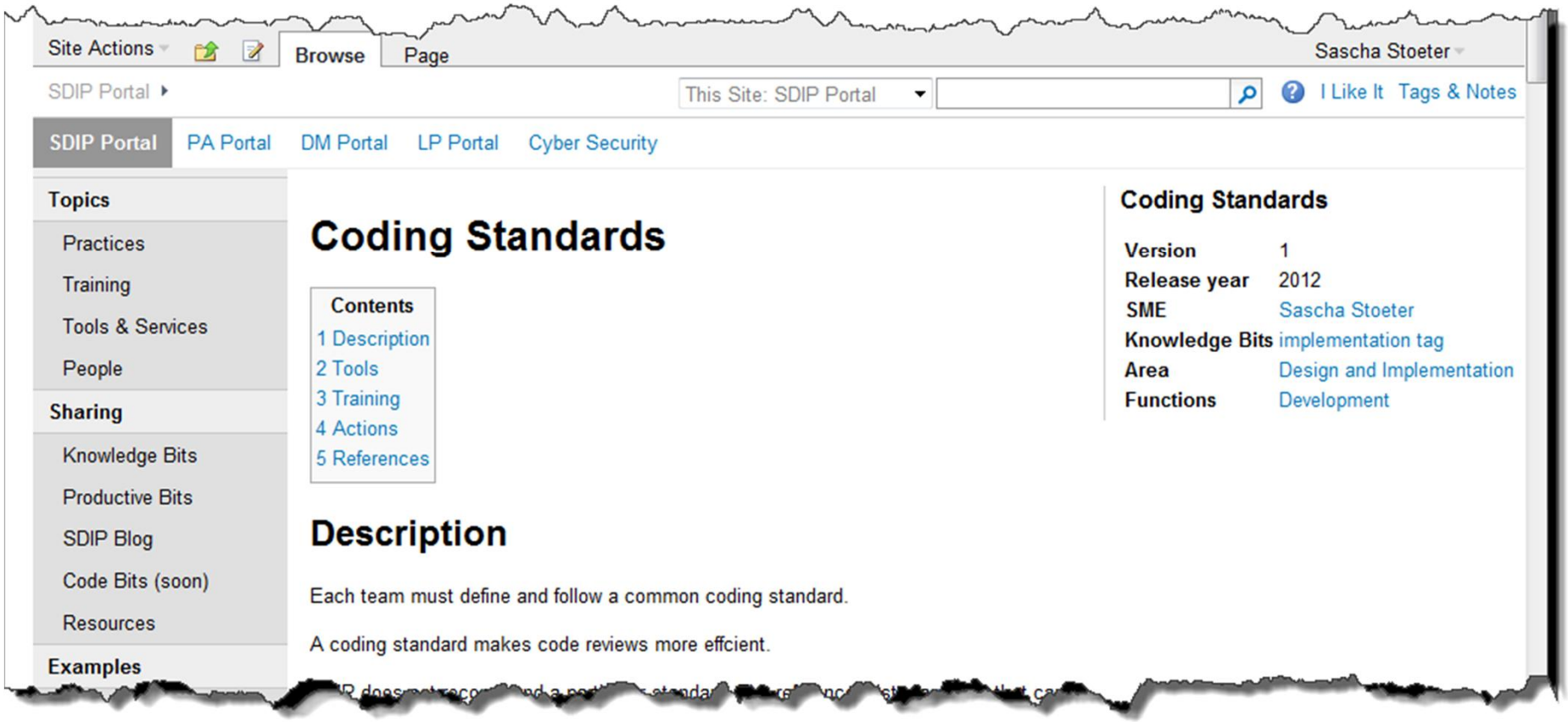
A practice describes the recommended way of executing an activity by joining people with a process description and tools. A typical practice is comprised of a description, available training, tools, SMEs, references and a link to a support community. It comes with an implementation plan and metrics to meet to demonstrate successful implementation. Depending on your organization's starting point, some practices can take a significant effort to be implemented. Others are as simple as picking a coding standard with no follow up actions.

Different practices affect different areas of the software development lifecycle and are not restricted to software developers. They touch on engineering, business decisions, project management and support. Likewise they are relevant for software engineers, product and line managers, architects and many more roles.

There are two sets of practices: mandatory practices that all teams must follow and a backlog of recommended, but optional practices. Occasionally practices are selected to become mandatory in accordance with the roadmap and the state of practice implementation throughout the company.

Learn more about individual practices, including the actions to take, by selecting an entry from the list.

Practice description



Static Code Analysis

What is Expected?	Status	Comment
Use Static Code Analysis on all new and changed code	↑	
For new and changed code, all warnings shall be analyzed and resolved	2	
Coding standards shall be enforced with the tool used	2	

What should be Measured?	Data Collected	Comment
The warning levels shall be monitored and controlled	1	

3 Ongoing	2 Ongoing	1 Ongoing	0 Not Started

↑ **Completed since last Baseline**

SDIP Site/Product L Report
<S duct Line>

Fill out Site or Product Line

Enter topics of interest to the Site

Update current quarter by changing colors

Hot Topic	Status	Comment
Organization and Resources		
Training Deployment		
Tool Chain Deployment		
2013 Planning and Budget		

Indicate status (Red, Green or Yellow of topics of choice and comment

Update current and projected status for practices (Green = Implemented, Red = To be implemented and Grey = Not applicable. (Note: No yellow!)

Indicate full implementation since last report using Green Arrow

Key Practice	Q1	Q2	Q3	Q4
Introduce Configuration			↑	
			2	3
			2	3
Code Review				
Coding Standards				
Unit Testing with Automation	1	1	3	
Project and Product Metrics	2	3	3	
Software Estimation				
R&D Self Assessments				
Nightly Builds				
Define and Control Interfaces				

Ongoing Activities

…

…

…

…

Concerns/Issues

…

…

…

…

List ongoing activities of choice and any concerns or issues

3 Ongoing	2 Ongoing	1 Ongoing	0 Not Started

↑ **Completed since last Baseline**

Communications and Sharing
Productive Bits

- New SDIP Blog created & Productive Bits Newsletter published bi-weekly
- More than two dozen locations have people printing it out and hanging it up in prominent locations in break rooms, common areas, etc.

Productive Bits Newsletter - Episode 1 - Drop It Cause It's Hot!

20. May 2012 20:21 - Brian P Robinson

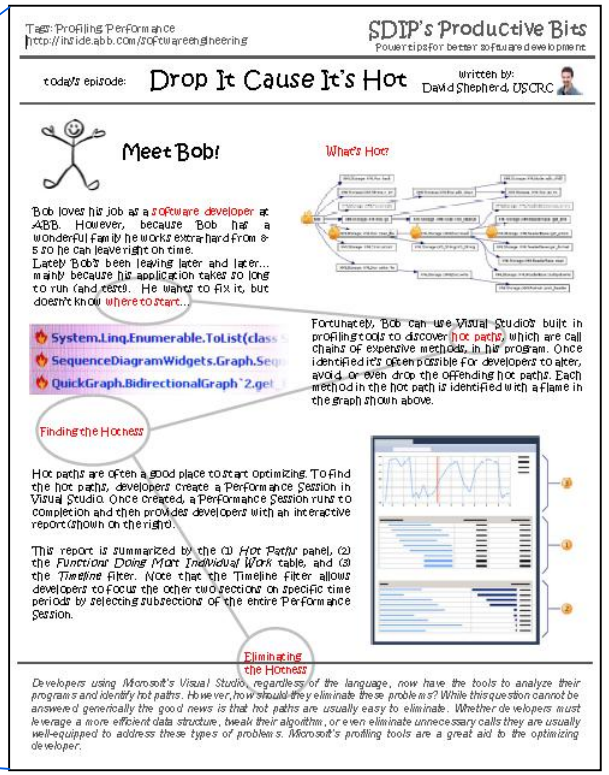
Introducing the new SDIP Productive Bits Newsletter, where software professionals inside ABB can share tips and tricks they use when developing software. These tips may involve productivity, quality, or any other topic of interest to the larger ABB development community. A new episode will be published every other week on this blog.

The first episode of the Productive Bits Newsletter is written erd, a Principal Scientist in the ISS Program at USCRC.

Introduction:
When a software project begins developers usually have one goal: make software that works. As the project moves forward their code soon begins working... and working and working and working. This brings them to their next goal: make software that works fast. However, at this point it's often difficult to determine which code segments are causing delays. Fortunately, developers can use Visual Studio's built in profiling tools to discover hot paths in their program. Once identified it's often possible for developers to alter, avoid, or even drop the offending hot paths.

Here is the newsletter:

Drop It Cause Its Hot - final.pdf



Tags: Profiling Performance
http://inside.abb.com/SOFtwareengineering

SDIP's Productive Bits
Power tips for better software development

today's episode: **Drop It Cause It's Hot**

written by: David Shepherd, USCRC

Meet Bob!

Bob loves his job as a software developer at ABB. However, because Bob has a wonderful family he works extra hard from 8-5 so he can leave right on time.
Lately Bob's been leaving later and later... mainly because his application takes so long to run (and crash). He wants to fix it, but doesn't know where to start...

What's Hot

Fortunately, Bob can use Visual Studio's built in profiling tools to discover hot paths, which are call chains of expensive methods in his program. Once identified it's often possible for developers to alter, avoid, or even drop the offending hot paths. Each method in the hot path is identified with a flame in the graph shown above.

Finding the Hotness

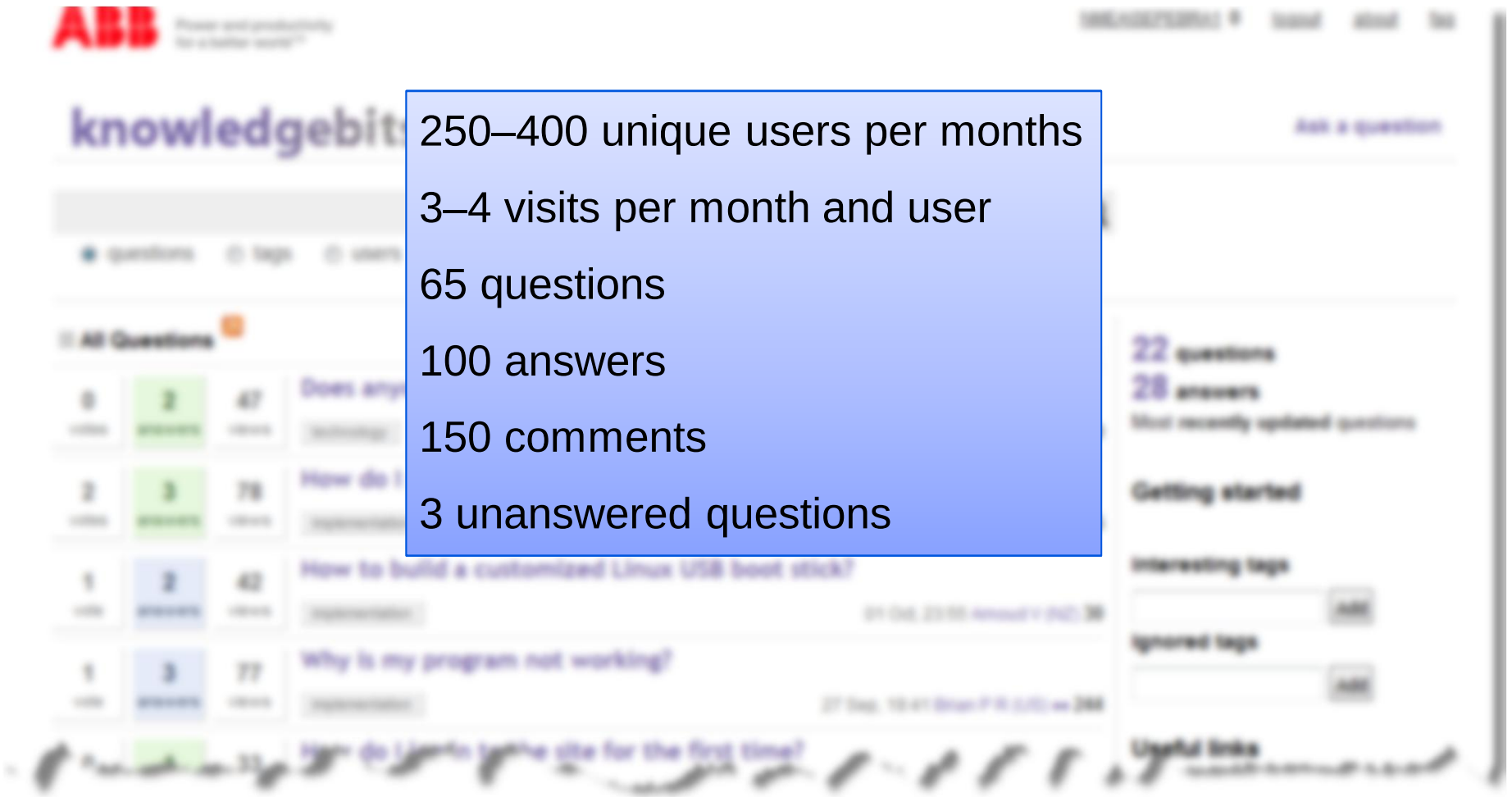
Hot paths are often a good place to start optimizing. To find the hot paths, developers create a Performance Session in Visual Studio. Once created, a Performance Session runs to completion and then provides developers with an interactive report shown on the right.

This report is summarized by the (1) *Hot Path* panel, (2) the *Functions Doing Most Individual Work* table, and (3) the *Timeline* filter. Note that the Timeline filter allows developers to focus the other two sections on specific time periods by selecting subsections of the entire Performance Session.

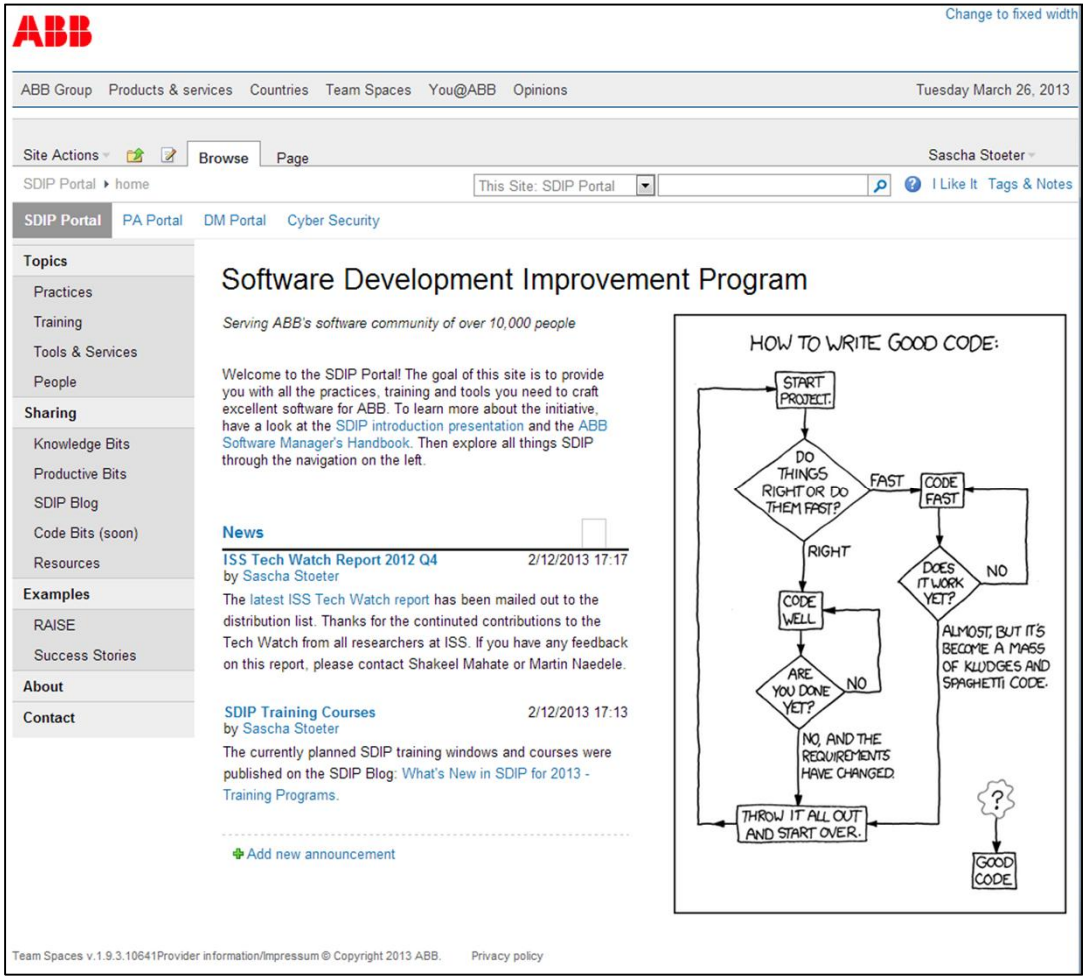
Eliminating the Hotness

Developers using Microsoft's Visual Studio, regardless of the language, now have the tools to analyze their programs and identify hot paths. However, how should they eliminate these problems? While this question cannot be answered generically the good news is that hot paths are usually easy to eliminate. Whether developers must leverage a more efficient data structure, tweak their algorithm, or even eliminate unnecessary calls they are usually well-equipped to address these types of problems. Microsoft's profiling tools are a great aid to the optimizing developer.

Communications and Sharing
Knowledge Bits



SDIP Portal
One stop shop for all things SDIP



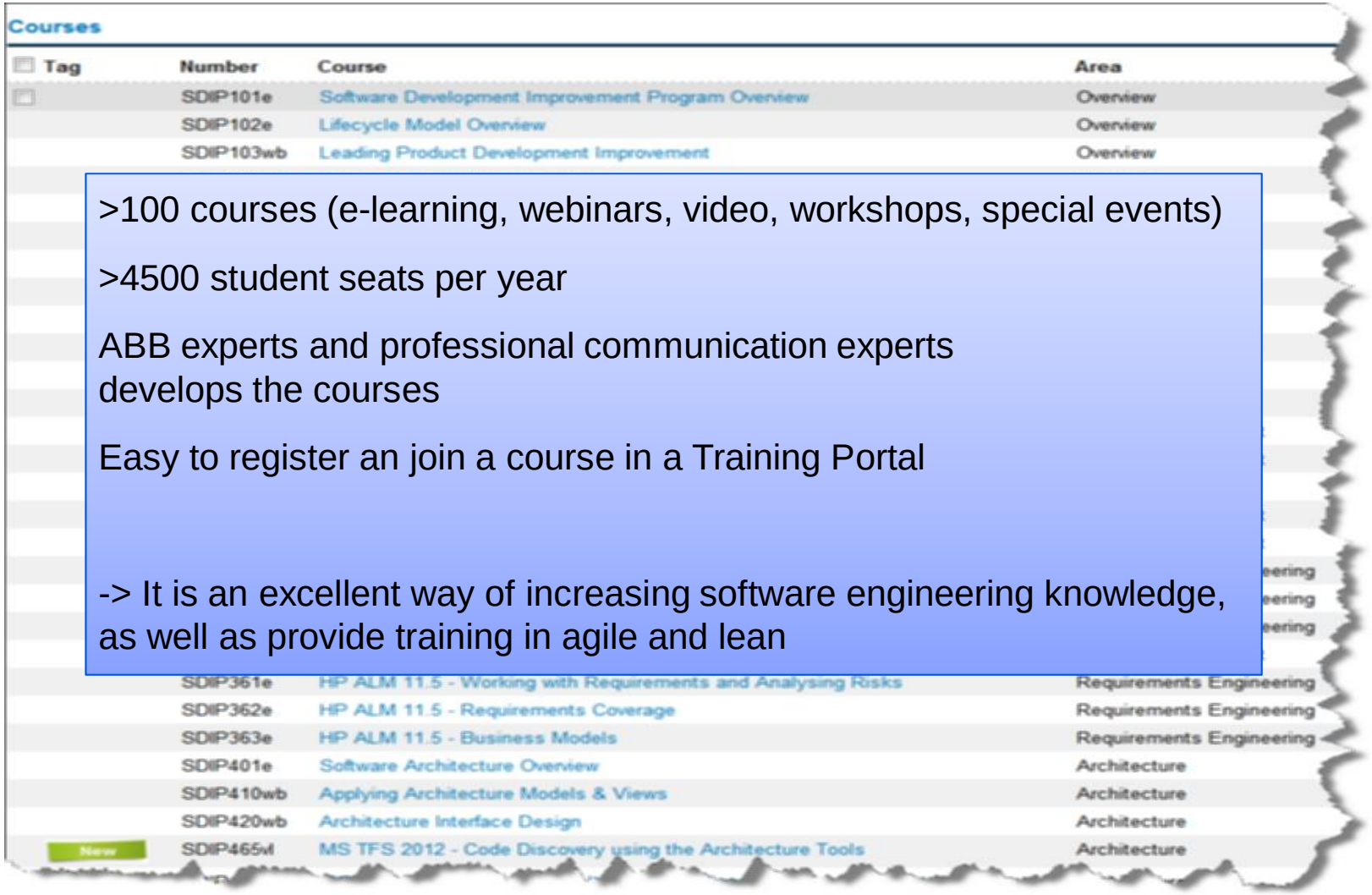
- Information consolidated in single site
- Cross-referenced information
  (practices, training, tools, people)
- Tailored views
  (eg, courses for managers)

SDIP Tools

- Convergence on single tool better than many better tools
  - Training
  - Cost

- SDIP selects and recommends tools
- Centrally administered tools
  - Installation, backups
  - Training and support
  - SME
  - Onboarding of teams
- Local power users

SDIP Trainings
Professional Training in Software Engineering



>100 courses (e-learning, webinars, video, workshops, special events)

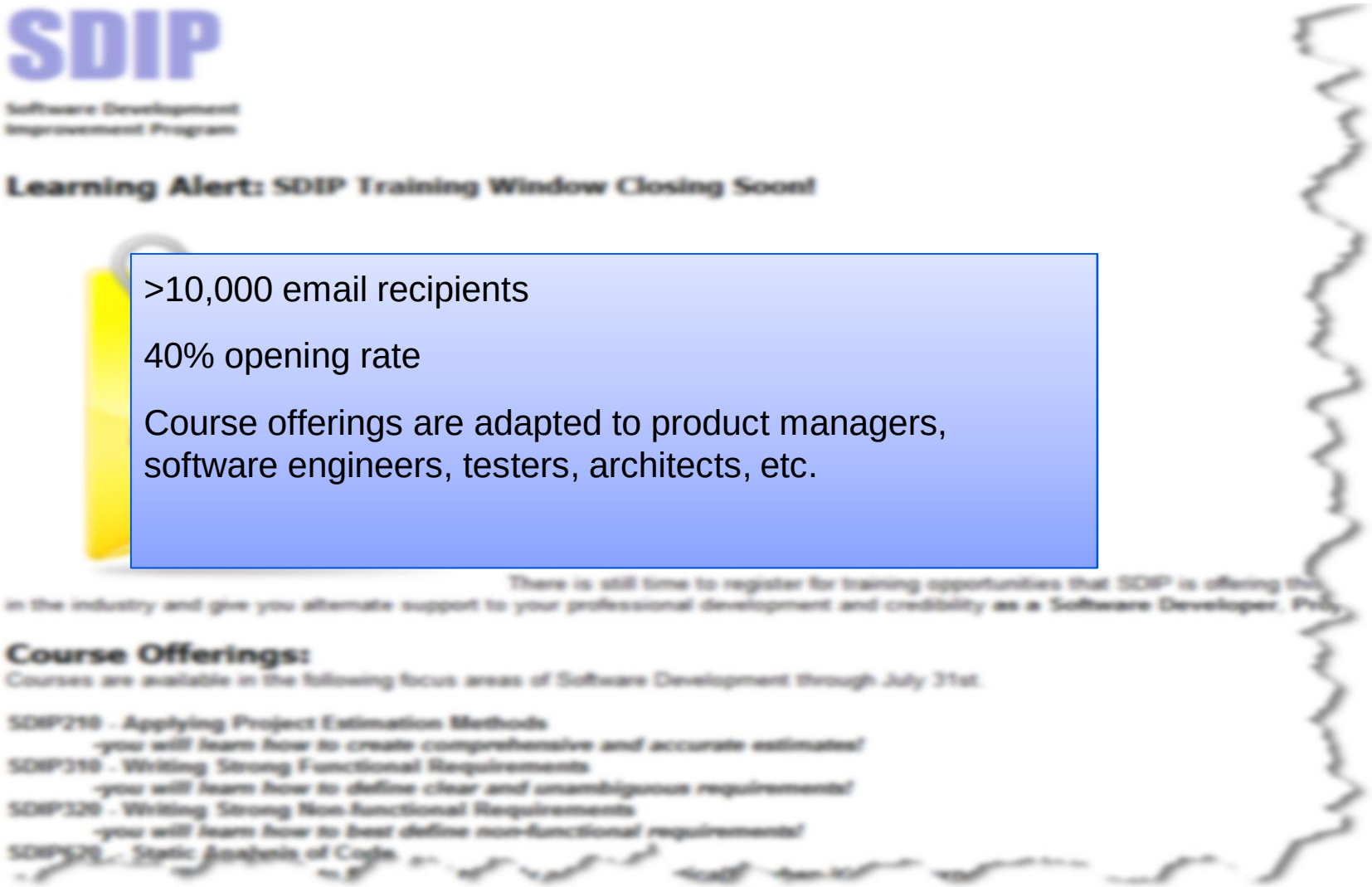
>4500 student seats per year

ABB experts and professional communication experts develops the courses

Easy to register an join a course in a Training Portal

-> It is an excellent way of increasing software engineering knowledge, as well as provide training in agile and lean

SDIP Trainings
Training Alerts to the Software Community


SDIP

Software Development Improvement Program

Learning Alert: SDIP Training Window Closing Soon!

>10,000 email recipients

40% opening rate

Course offerings are adapted to product managers, software engineers, testers, architects, etc.

There is still time to register for training opportunities that SDIP is offering this ... in the industry and give you alternate support to your professional development and credibility as a Software Developer. Pro...

Course Offerings:

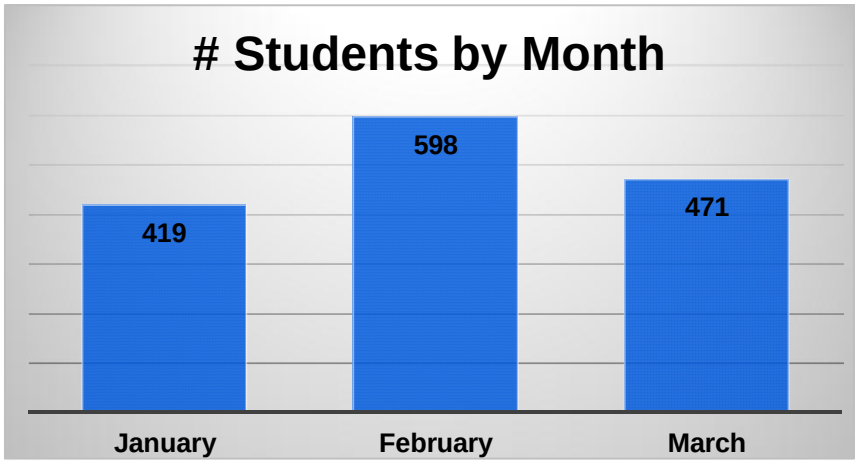
Courses are available in the following focus areas of Software Development through July 31st.

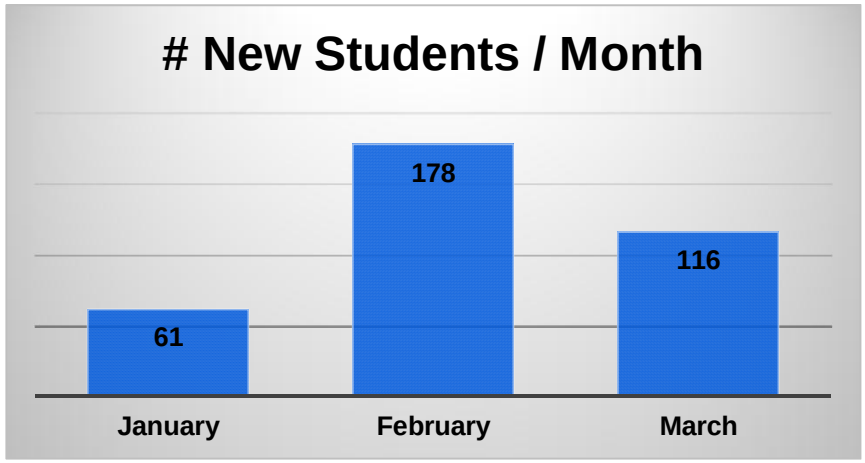
SDIP210 - Applying Project Estimation Methods
-you will learn how to create comprehensive and accurate estimates!
SDIP310 - Writing Strong Functional Requirements
-you will learn how to define clear and unambiguous requirements!
SDIP320 - Writing Strong Non-functional Requirements
-you will learn how to best define non-functional requirements!
SDIP370 - Static Analysis of Code

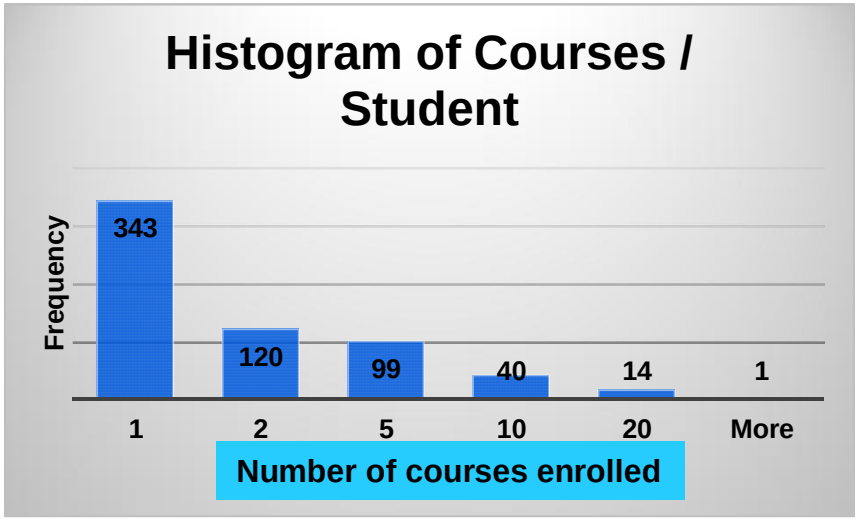
SDIP Training Program
Training Windows 2013

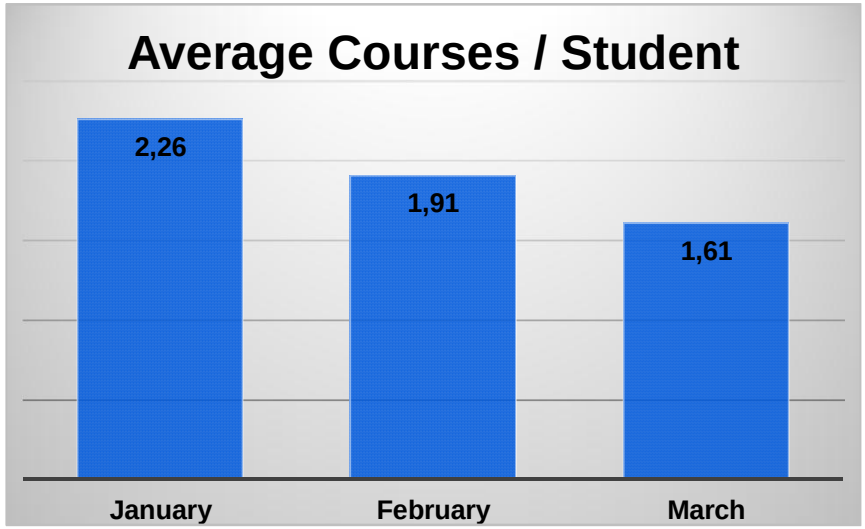
- Initial Promo (all self paced courses): Jan+, *sent Jan 25th*
  - New: HP ALM 11.5 e-learnings, Klocwork instructional videos
  - Black Duck e-learnings, HP Service Manager
- 1st Window: Feb/March – *promo Feb 4, classes Feb 18-Mar 28*
  - New: SCRUM Software Development
  - Existing: Requirements, Architecture
- 2nd Window: April/May – *promo April 2, classes April 16-May 23*
  - New: Leading Product Development Improvement, Business Scenarios & User Stories
  - Existing: Unit Testing, Quality Assurance, Static Analysis
- 3rd Window: …

SDIP Training Program
Global Participation Statistics – 2013 Year to Date









SDIP Training Program
Participation by Division – 2013 Students to Date

Division	Total Students	Percent of Total
PA - Process Automation	468	39%
PS - Power Systems	415	34%
DM - Discrete Automation & Motion	187	15%
PP - Power Products	96	8%
LP - Low Voltage Products	45	4%
ZC - Corporate & Other	276	NA

Total Students = 1,487

SDIP Training Program
Participation by Country – 2013 Year to Date

United States	394
India	278
Sweden	134
Switzerland	110
Germany	104
Poland	85
China	63
Finland	62
Unknown	40
Norway	39
Italy	35
Canada	33
France	18
Australia	18
Ireland	9
Estonia	9
Czech Republic	7

Singapore	7
Croatia	7
Brazil	6
Spain	6
Netherlands	5
United Kingdom	4
Mexico	3
Russian Federation	2
Korea, Republic of	1
Colombia	1
Denmark	1
Saudi Arabia	1
Oman	1
Vietnam	1
Chile	1
Qatar	1
Japan	1

SDIP Training Program
Student Evaluation Results – 2012 Yearly Totals

Evaluation Question	Very Good	Good	Neutral	Poor	Very Poor
Overall how would you rate the content of this course?	19%	55%	22%	4%	0%

Evaluation Question	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
The instructors of this course were knowledgeable?	36%	54%	7%	2%	1%
The instructors' had a positive impact on my understanding of the material?	26%	55%	16%	2%	1%
The Webinar content was well organized?	28%	59%	11%	2%	0%
Provided the knowledge necessary to achieve the objectives of the course?	20%	58%	18%	4%	0%
The Webinar content was relevant to my position at ABB?	40%	45%	13%	2%	0%

Note: based on 419 student responses

← Overall student satisfaction →
95%

SDIP Training Program
Student Evaluation Results – 2012 Yearly Totals (con't)

Evaluation Question	Way Too Fast	A Bit Too Fast	Just Right	A Bit Too Slow	Way Too Slow
Overall; how would you rate the pace of this course?	2%	8%	73%	14%	3%

Evaluation Question	Not Enough	Just Right	Too Much
Overall; how would you rate the level of interaction in this course	10%	87%	3%

Evaluation Question	Too Few	Just Right	Too Many
Overall; how would you rate the number of examples in this course	7%	85%	8%

Which topic or element in the Webinar do you feel will be most useful to your work at ABB?

Answer	Number	Percent
General course content	182	45%
Instructor led examples	139	34%
Participant exercises	86	21%

Summary

- Achievements
  - Traction and recognition in company
- Challenges/Goals
  - Building a community
  - Making ABB a known and desired workplace place for software engineers
- Lessons learned
  - Small core team can have an impact in a large organization
  - Carrot wins over stick (but peer pressure helps)
- Open question
  - Potential improvements through disruptive organizational changes


ABB