

Talking to managers about modernizing .NET apps

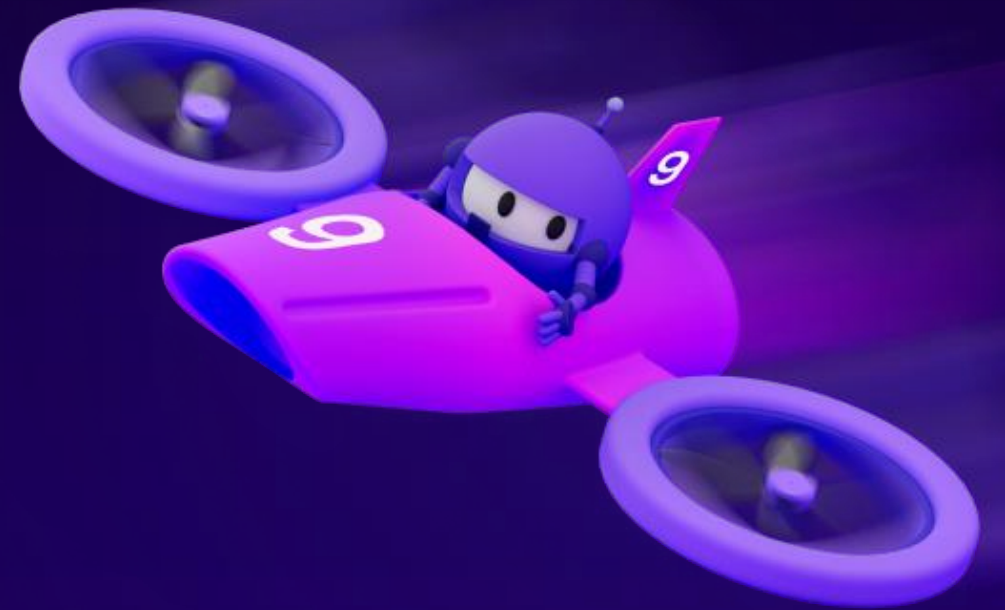
Tomas Herceg

Microsoft MVP

CEO @ RIGANTI

Author of DotVVM

tomasherceg.com



Agenda

01

**Describe
risks & benefits**

02

**Manage
expectations**

03

**Communicate
the progress**

Problems of .NET Framework apps

Security

Old & unsupported libraries

- Probably vulnerable

Old authentication methods

- Forms Authentication
- On-premises Active Directory

Weak password hashing

- ASP.NET Membership Providers
- ASP.NET Identity 1 or 2

Difficult to not have secrets in code

Frameworks were not always safe by design

- Explicit HTML encoding in ASP.NET Web Forms
- `<%# ... %>` vs `<%#: ... #>`

Reliance on machine key

Problems of .NET Framework apps

Performance, operating & maintenance costs

Windows-only

- Heavy when running in containers
- BYOD problems

Higher operating costs in on-premises and cloud

- You need more servers
- Bing team reported **34% improvement** when going from .NET Framework 4.7.2 to **.NET Core 2.1**
- Another **40-50%** gained by moving to **.NET 6**

Monolithic design & lack of modularity

- Modernization is hard in monoliths
- System-wide installation of .NET Framework

Hundreds of tiny obstacles = more dev hours

- Libraries cannot rely on common infrastructure (logging, configuration model, dependency injection...)
- More Git conflicts in project files
- Difficult testability
- Binding redirects
- ...

Problems of .NET Framework apps

Hiring & developer satisfaction

Increasingly difficult to find developers

- .NET Core was released 8 years ago
- Younger .NET developers know only the new stuff

Unattractive for new hires

Impossible to stay up-to-date with knowledge

- Developers cannot experiment with new libraries & features
- Their value in the job market will decrease

Inferior tooling and missing capabilities

- Visual Studio & .NET bring of improvements
- Most of them do not work in old projects

Agenda

01

**Describe
risks & benefits**

02

**Manage
expectations**

03

**Communicate
the progress**



**Every modernization
project is unique**

Evaluate & experiment

Total rewrite

You can change everything

- Architecture & approach
- Libraries & frameworks
- Feature set

Hard to reuse code

Difficult transfer of know-how

- You often need a second team

Incremental modernization

Migrating pages / features / other small blocks

Impossible to make big changes

- Most concepts must remain until everything is migrated

Most code can be reused

- Numerous, but trivial edits

Retreat to previous version is possible

- Feature-toggles, A/B testing

Making estimates

Count & classify

ASP.NET Web Forms or MVC pages

- How many of them
- Complexity (e.g. Small, Medium, Large)
- User controls can be treated as separate pages

ASP.NET Web Services, WCF, ASP.NET Web API

- How many endpoints
- Complexity of input & output models

Entity Framework 6 to EF Core

- How many LINQ queries on DbContext
- How many references to lazy-loaded properties

ASP.NET Membership or old ASP.NET Identity

Other .NET Framework features & libraries

Making estimates

Experiment – try to migrate few instances of each class

Item	Count	Migrated items	Effort (hrs)	Estimate (hrs)
Web Forms – Small page	16	3	5.7	30.4
Web Forms – Medium page	39	3	19.2	249.6
Web Forms – Large page	18	3	45.7	274.2
Entity Framework queries	52	5	8.3	86.3

Add other migration items with constant effort

- Migration to ASP.NET Core authentication packages
- Migration to the new configuration model
- Migration of ASP.NET Membership Providers to ASP.NET Core identity...
- Creating CI/CD pipelines for the new project
- ...

Prepare “business plan”

Document, presentation, ...

Risks of NOT DOING the modernization

Benefits

- Financial & non-financial

Effort estimate

- How many team members
- Expected distribution between modernization & maintaining the current version
- Emphasize that first estimate is never accurate

Risks of DOING the modernization

- Every modernization project is unique
- Complications are likely to happen

How will the process look like

- The estimate will be updated regularly based on actual progress

Agenda

01

**Describe
risks & benefits**

02

**Manage
expectations**

03

**Communicate
the progress**

Communicate the progress

Predictability is worth more than on-time delivery

Update the table with estimates regularly

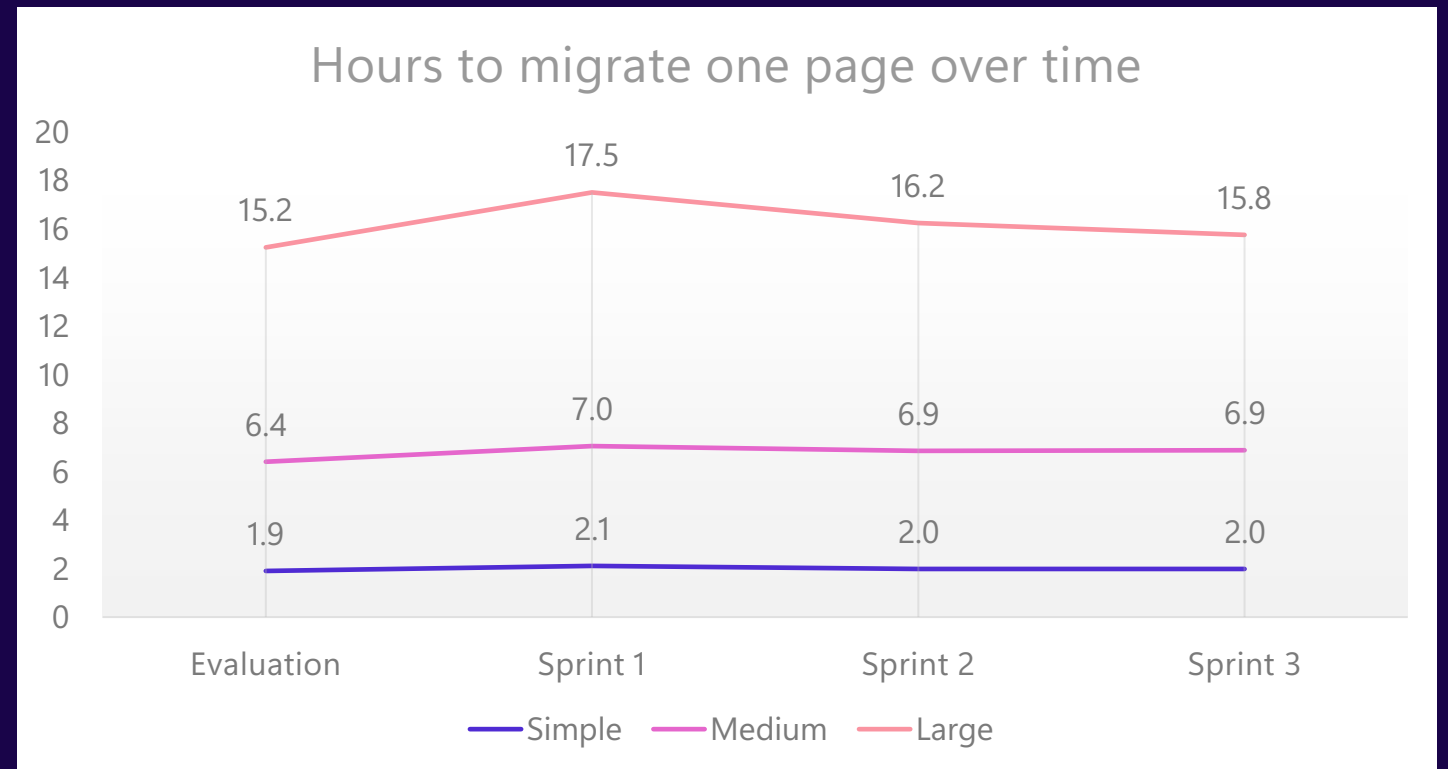
- Track time spent on individual items
- Share the data with the team

Stop & brainstorm when you get off the track

- Explain large deviations from the plan
- Be prepared to change the strategy

Presentations & reports are not waste of time

- It's better to know about problems in advance



Learn more

ASP.NET Web Forms

ASP.NET MVC

ASP.NET XML Web Services

WCF

ASP.NET Web API

SignalR

LINQ to SQL

Entity Framework

ASP.NET Membership Providers

...

modernizationbook.com



Modernizing .NET Web Applications

Everything You Need to Know About
Migrating ASP.NET Web Applications
to the Latest Version of .NET

—
Tomáš Herceg

Apress®

Q&A

