Core development, Embedded development

Team CoreDev
Contents:

1. Portfolio
2. Fields
   a. Focus on Engineering
      i. Case study characteristic curve calculation
      ii. Case study embedded development
      iii. Case study converting of CAD data
3. How we work
   a. Team
   b. Adaptability
   c. Quality and development process
   d. Source code access through TFS possible
   e. Adaptability and Quality Assurance
Portfolio
What we do ...

- Supporting physical modelling
- Supporting mathematical modelling
- Graph modelling
- Calculation design
- Algorithm design
- Architecture
- Programming
- Gfx and Shader programming
- Documentation
... on these platforms ...

- Windows 32/64 Native
- .NET Framework (for Windows)
- .NET Core (for Windows, Linux and Apple, also for servers)
- Smartphone (XAMARIN)
- Embedded (with or without operating system)
... for these types of applications ...

- Applications with a simple UI
- Applications with a complex UI (assisted by TeamX, A-Team or external developers)
- Plugins
- Command line applications
- System Services
- REST Web services
- Embedded Development
... with these programming languages

- C/C++
- C#
- Java
Fields
Our focus is on Engineering

We are offering software services in desktop and embedded environments, focusing on the following fields:

- **Mechanical Engineering**
  - CAD
  - Computation of characteristics
  - Simulation
- **Automotive Engineering**
  - Gear units, gearboxes, clutches
  - Electric drives
  - Software development for control devices
Our focus is on Engineering

- **Measurement Engineering**
  - Data analysis
  - Modelling

- **Electrical Engineering and Mechatronics**
  - Calculations
  - Characteristics
  - Embedded development

- **Calculation, Modelling and Simulation**
  - Characteristics
  - Material stress
Our focus is on Engineering

- CAD and Design engineering
  - Conversion
  - Data enrichment
  - Visualization

- ODE problem solving with numerical or analytical methods
  - ODE = ordinary differential equation
Case study
Calculation of characteristics
Calculation of characteristics - The Requirements

It is required to generate a series of characteristics for a complex mechanical component, from which the physical and mechanical behavior of the component can be predicted.
A mathematical model is developed with the assistance of mechanical engineers to predict part behavior. The model can be specified in various manners, for example via Simulink, Excel or simply as a set of equations in a document.
The implementation can be developed with classical or agil development methods, depending on project or client needs.

Depending on the complexity of the implementation and the wishes of the customer, results can be shown at an early development state to validate software features.

Development states:

**klassisch**
- Prototype
- Alpha
- Beta
- Release Candidate
- Release

**agil**
- MVP
- Iterative improvements through stakeholder feedback after each sprint
Case study
Embedded Development
An existing software/firmware of a measuring device has to be enhanced with more features and better runtime stability.

The software has been developed by an unexperienced programmer and thus is very unstable and hard to read.

The customer wants some features to be added and existing bugs to be fixed.

Development language was Standard C.

The Customer has provided us with one device for testing and evaluation.
Embedded development - Preparations

The developers encounters an unfamiliar development environment, which has to be prepared in advance, in order to be enabled to edit, compile, and debug the program code.

In the course of preparation, the existing code has to be analyzed and assessed.

The main outcome of the preparation phase is an estimation of the development effort needed to fulfil the task.
Embedded development - possible hazards

Working with the program code of an unexperienced developer in an unfamiliar environment can be a risky enterprise. The fact, that we are dealing with a cross platform environment in our example, adds even more risk to all involved parties.

The real development time can turn out to be a multiple of the original estimation.

Yet undiscovered lurking surprises within the original source code can rapidly inflate the development time by factor 3 or more.
Selective surgical corrections in the existing code turn out to be impossible in our given example. The existing software could not be extended.

It is therefore decided to completely rework the existing code, which, after all, turns out to be an almost complete redevelopment. Additionally, a Windows simulation is developed in order to enable the developer to debug effectively, which will save a lot of time in the overall development process.

Reauthoring the software creates a solid foundation on which future developments can be built. Future development obstacles can be eliminated by getting rid of the relics in the original code.
Case Study
Processing of CAD data
Processing of CAD Data - Specification

Volumetric and mesh data, generated by a CAD system, is post processed in a time consuming manual task. The post processing task is necessary for downstream processing, like for example production.

The data format of the CAD system is proprietary, but the customer has a specification of the file format. The CAD system is furthermore plugin capable and is featured with an XML export function.

It a requirement, that the manual task has to be automated.

This leads to the following possibilities of implementation:

- A converter, which consumes the proprietary format and generates a new file with the desired result
- A converter, which consumes the XML-file and generates a new file with the desired result
- A plugin, which directly exports the target file
After a discussion of the pros and cons of all possibilities, it has been decided, that the plugin solution is the most suitable solution.

After a development time of several hundreds of man hours, an elegant solution can be supplied, which saves, the staff for our customer, a huge amount of time.

The costs of development will pay for itself within a couple of months, with one positive side effect. The motivation of the employees is significantly increased.
Principles of development
Principles of development of team*Σ

- team*Σ combines classical Milestone development with agile elements:
  - The project documentation grows with the progress of the project
  - The project can be specified by ...
    - .... the users point of view
    - .... a result of mathematical/physical modelling
    - .... as a roughly blueprinted specification, yet to be fine tuned
  - The collaboration with other K&K Software AG Teams, like the A-Team or TeamX is as well possible, as is cooperation with external developers or development teams
- On site development at the customers location is possible in special cases
- Close collaboration with the experts of the target field is possible and necessary
- The implementation process is mainly autonomous
**Principles of development: Flexibility**

- Minor requested changes can be processed without the increase of budget, if the increase of man hours will not exceed a certain limit
- Development in one or more of these programming languages is possible:
  - C/C++
  - C#
  - Java
- Modular multi platform development is possible for:
  - Windows
  - Linux
  - Apple
- Modules can be realized as ‘Ready For Web’ modules
Principles of development: Quality and Phases

- QA starts with planning:
  - Top Down planning of the software
  - Bottom Up implementation of the software
  - QA by Unit tests when applicable and meaningful

- Utilization of classical and proven project phases:
  - Prototype
  - Alpha
  - Beta
  - RC
  - Release

- Sub Milestones and partial deliveries / partial releases are possible and in some cases customary.
Source code access via TFS

The customer is enabled to control the development state on a steady base. This makes the development process transparent.

You can get access to the source code, the project tree and of course the backlog of our TFS server.
Conclusion

- Flexible development on various platforms:
  - Windows/Linux Native (only C/C++)
  - Embedded (C/C++)
  - .NET Framework / .NET Core (C++ or C#)
  - Java VM (Java)
  - .NET Framework managed code and native code can be mixed

- Quality assurance
  - Module based testing (is necessary and meaningful) with Unit Tests
  - Automated case testing
  - User based testing can be offered
  - Comprehensible Documentation
Any questions? Please contact us!

We are committed to your request and will be happy to answer detailed questions promptly. We look forward to your inquiry.

I am your contact person:

Laura Köpl  
*Head of Marketing and Sales*

Phone: 09382 / 3102-241  
Mail: koepl@kk-software.de
Protect your interests:

Develop excellent products with our team at K&K Software AG, now.