SW AS A CAPITAL
DEVELOP REUSABLE SW FROM THE BEGINNING

HISAO MUNAKATA
SENIOR DIRECTOR, HPC MARKETING DIGITAL DIVISION

DAN SISCO
SENIOR DIRECTOR, R-CAR GEN5 PROGRAM MANAGER

HIGH PERFORMANCE COMPUTING, ANALOG AND POWER SOLUTIONS GROUP
RENEAS ELECTRONICS CORPORATION
## TRANSITION OF INNOVATION SOURCE
### HOW SW BECAME THE CENTER OF TECHNOLOGICAL GRAVITY

<table>
<thead>
<tr>
<th>Core competitiveness</th>
<th>In the past</th>
<th>Recently</th>
<th>In the future</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology owner</td>
<td>PCB circuitry</td>
<td>SoC</td>
<td>SW</td>
</tr>
<tr>
<td>Key demand</td>
<td>robustness</td>
<td>integration</td>
<td>portability</td>
</tr>
<tr>
<td>Image</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
<tr>
<td>Challenges</td>
<td>ECU interconnect, age-related fault</td>
<td>Performance, power (thermal)</td>
<td>SW update, cyber security</td>
</tr>
</tbody>
</table>

Now, many OEMs claim SW has become a primary source of product competitiveness (= SW first concept)
STILL, ONE SIZE (SW) DOES NOT FIT ALL
EMBEDDED SYSTEMS HARDLY DECOUPLE HW (SOC) AND SW

HW variation
- Require performance (=price) variants
- Expect unique competitive features (display numbers, AI,..)

SW commonality
- Production variants (same generation)
- SW migration (across the generation)

HW/SW boundary
- Application portability (OS interface)
- HW access interface (common API)
- System level boundary

Generic System Images (GSIs)
A Generic System Image (GSI) is a pure Android implementation with unmodified Android Open Source Project (AOSP) code, runnable on a variety of Android devices.

App developers can install and run the latest Android GSIs to perform app testing on a variety of existing Android devices and using GSIs from different Android OS release stages, including Developer Preview and Beta builds. Adding GSIs to your verification and testing processes can provide you with some extra benefits:
- Broader test coverage on a greater set of real devices
- More time to fix app compatibility issues
- More opportunities to fix compatibility issues in Android that are reported by app developers

The GSI project is open source and helps improve the Android ecosystem by providing more ways to improve app and OS quality before each release of Android.

Android mandates handset vendors to support GSI boot, a common binary that can boot multiple devices.
EXPECTED SW REUSE CANDIDATES

... AND VARIOUS BLOCKING FACTORS FOR EMBEDDED SW REUSE

① Same OEM, But different Tier1

② Same Tier1, different OEM

③ Same Tier1, different team

④ Same Tier1, Different generation

⑤ SW reuse across different OEMs (open industry collaboration)
## VARIOUS BLOCKING FACTORS FOR SW REUSE

Not only HW-oriented, but also SW-related and business-related

<table>
<thead>
<tr>
<th>HW origin</th>
<th>SW origin</th>
<th>Business origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>Data definition</td>
<td>Contract</td>
</tr>
<tr>
<td>• Speed</td>
<td>• Common language</td>
<td>• Code disclosure</td>
</tr>
<tr>
<td>• Memory size</td>
<td>• Coding convention</td>
<td>• Asset ownership</td>
</tr>
<tr>
<td>Major IP incompatibility</td>
<td>• MISRA-C</td>
<td>License</td>
</tr>
<tr>
<td>• CPU / GPU / NPU</td>
<td>• Code management</td>
<td>• Copyright restriction</td>
</tr>
<tr>
<td>Minor IP incompatibility</td>
<td>• Repository access</td>
<td>• Warranty scope</td>
</tr>
<tr>
<td>• Register function assign</td>
<td>• Readability</td>
<td>Deliverables</td>
</tr>
<tr>
<td>• Channel number</td>
<td>• Document density</td>
<td>• Source code access</td>
</tr>
<tr>
<td>Environment</td>
<td>• Language</td>
<td></td>
</tr>
<tr>
<td>• IDE, debug board,..</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Environment</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SW AS A CAPITAL, NOT AN ASSET
CAPITAL SHOULD BE RESourced TO GENERATE REPEATED FUTURE VALUE

SW as an Asset (traditional way)
- Acquired to date
- Accumulated over time
- Care for the preservation of assets
- The value diminishes over time
- Hard to maintain the present value
- Becomes a non-performing asset

Collection of past SW code became bad debt

SW as a Capital (new direction)
- Source of future value
- Core of the growth strategy
- To increase the prospective value, you need to reinvest in the SW
- SW management strategy should include “What exactly to do for expansion and reinvestment?”
“CAPITAL TURNOVER RATE” GIVES PERFORMANCE INDEX
IT SHOWS HOW MUCH REVENUE GENERATED FROM THE EXISTING ASSET

Capital Turnover Rate (Accounting terms)

- **Capital Turnover Rate** = Annual sales / stockholder equity (net worth)
  - The ratio indicates how much a company could grow its current capital investment level.
  - Its value varies by type of business. (1.3 ~ 3.5)

- **SW Capital turnover Rate** = Reused code / Total developed code
  - We can calculate the true code reuse rate.
  - That does not include the use of vague experience and know-how.
  - We need to consider how we can improve the true code reuse ratio.
HOW TO REALIZE CAPITALIZED SW (= REUSABLE SW)
"REUSE" IS ESSENTIAL FOR SW CAPITALIZATION

<table>
<thead>
<tr>
<th>Eliminate Assumptions</th>
<th>Follow Coding Conventions</th>
<th>Interoperability with other industries</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Do not optimize the code</td>
<td>● Style format (tab length, codepage,..)</td>
<td>● Cloud connectivity</td>
</tr>
<tr>
<td>Optimize may include assumption</td>
<td>● Pursuit of readability (doxygen support,..)</td>
<td>● Mobile network connectivity</td>
</tr>
<tr>
<td>● HW configuration (ch. number)</td>
<td>● Nomenclature (adopt common naming)</td>
<td>● Vehicle service connectivity</td>
</tr>
<tr>
<td>● Data range</td>
<td>● Use code control system (proper use of git system)</td>
<td>● common control vocabulary (shared semantics) for data sharing/integration</td>
</tr>
<tr>
<td>● Memory size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>● Execution speed (network,..)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# WHAT COVESA CONTRIBUTES TO REALIZE CAPITALIZED SW

COVESA HOSTS VARIOUS COLLABORATIVE PROJECTS THAT PROMOTE SW CAPITALIZATION

<table>
<thead>
<tr>
<th>Data Expert Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Data Models &amp; Science</td>
</tr>
<tr>
<td>● VSS eco-system</td>
</tr>
<tr>
<td>● Common Vehicle Interfaces</td>
</tr>
<tr>
<td>● VehicleAPI (Autosar)</td>
</tr>
<tr>
<td>● Service definitions</td>
</tr>
<tr>
<td>● VISS</td>
</tr>
<tr>
<td>● IFEX IDL framework</td>
</tr>
<tr>
<td>● Architecture / Infrastructure</td>
</tr>
<tr>
<td>● Data Layer</td>
</tr>
<tr>
<td>● Data Centric Arch</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Automotive AOSP App Framework Standardization Expert Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Non-GAS emulator</td>
</tr>
<tr>
<td>● Data/Sensor API Definition</td>
</tr>
<tr>
<td>● e.g. camera</td>
</tr>
<tr>
<td>● Push Notifications</td>
</tr>
<tr>
<td>● Login &amp; Identification</td>
</tr>
<tr>
<td>● Payment</td>
</tr>
<tr>
<td>● Location Based Services</td>
</tr>
<tr>
<td>● Driver distraction</td>
</tr>
<tr>
<td>● Display management</td>
</tr>
<tr>
<td>● Touchpad support</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Electric Vehicle Charging Expert Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>● EV Charging Event Data Aggregation</td>
</tr>
<tr>
<td>● EV Optimization</td>
</tr>
<tr>
<td>● Private Cross OEM Joint Compute for Changing</td>
</tr>
<tr>
<td>● Delivering great EV experiences</td>
</tr>
</tbody>
</table>
TRANSLATING SW CAPITALIZATION TO RENESAS DEVELOPMENT PILLARS TO SUPPORT THE TRANSFORMATION

Create SW Re-investment Strategy

Eliminate Assumptions

 Improve Industry Collaboration

Single Mainline Dev’t

Cloud Native Dev’t

Simulation Enabled Dev’t

Increased Standardization via Consortium and OSS Leadership and Participation
REnesas Pillars to Pursue SW Capitalization
Single Mainline Development

- “Single mainline” development co-locating and integrating all SW features at right
- Increased chipset and vehicle system integration planning and test
  - better visibility into SW re-use in real world cases
- Optimizations and customizations become branches from here

Make smart optimizations to avoid jeopardizing SW re-useability
RENESAS PILLARS TO PURSUE SW CAPITALIZATION
CLOUD NATIVE DEVELOPMENT THROUGHOUT VEHICLE LIFETIME

Cloud Native Dev’t

- “One stop shop”: Single point of entry and operation for complete SW dev’t workload
- Product Simulation (fast/medium/slow)
- AI analysis/simulation
- Documentation/SW access
- Compute rental mechanisms (HiL, SiL)

Cloud Native to pull Dev’t Left and Enable Broader and Easier SW Access
RENESAS PILLARS TO PURSUE SW CAPITALIZATION
SIMULATION ENABLEMENT FROM THE START

**Simulation Enabled Dev’t**

- Renesas Automotive SW IDE includes simulation components since 2021
  - Multi-device simulation
  - Enhanced debug/trace
- Now we expand on this
  - Flexibility of simulation (speed/accuracy)
  - Deeper AI toolchain simulation

*Functional development and validation*

**Virtual SP**
- Ref app (DMS, Front camera and SV)
- MW
- MDFWK
- GPC FWK
- CV/Video pipeline
- OSAL
- MCAL
- BSP

**Simulator**
- Application domain (ARM CA cores)
- Real domain (ARM CR cores)
- GP-DSP
- AI Acc
- GPGPU
- ISP
- IMR
- Video I/O
- Peripheral I/O
- FPGA instance PC/Cloud
- Host machine PC/Cloud

**Performance validation**

- Ref app
- MW/BSP/MCAL
- HW access patterns, data and sequence

*Decouple SW Value and Strategy from HW Access*
Renesas is making great efforts to **master our own journey towards SW capitalization**, and to **enable our customers and partners** to master their own journeys…

**OUR JOURNEY…**

**GENERAL METHODS**
- Eliminate Assumptions
- Follow Coding Conventions
- Interoperability with other industries

**RENESAS APPROACH**
- Single Mainline Dev’t
- Cloud Native Dev’t
- Simulation Enabled Dev’t

**SW as a *Capital* (new direction)**