

# Debugging .NET and Native Applications in the Field



**Gad J. Meir**  
**IDAG** Ltd.

**Bug Exterminator & Process Plumber**

**EBlog:** [weblogs.asp.net/gadim](http://weblogs.asp.net/gadim)  
**HBlog:** [blogs.microsoft.co.il/blogs/gadim](http://blogs.microsoft.co.il/blogs/gadim)  
**Email:** [gadim@idag.co.il](mailto:gadim@idag.co.il), **Site:** [www.idag.co.il](http://www.idag.co.il)

**2011-11-01dotnet ruhrpott.net**

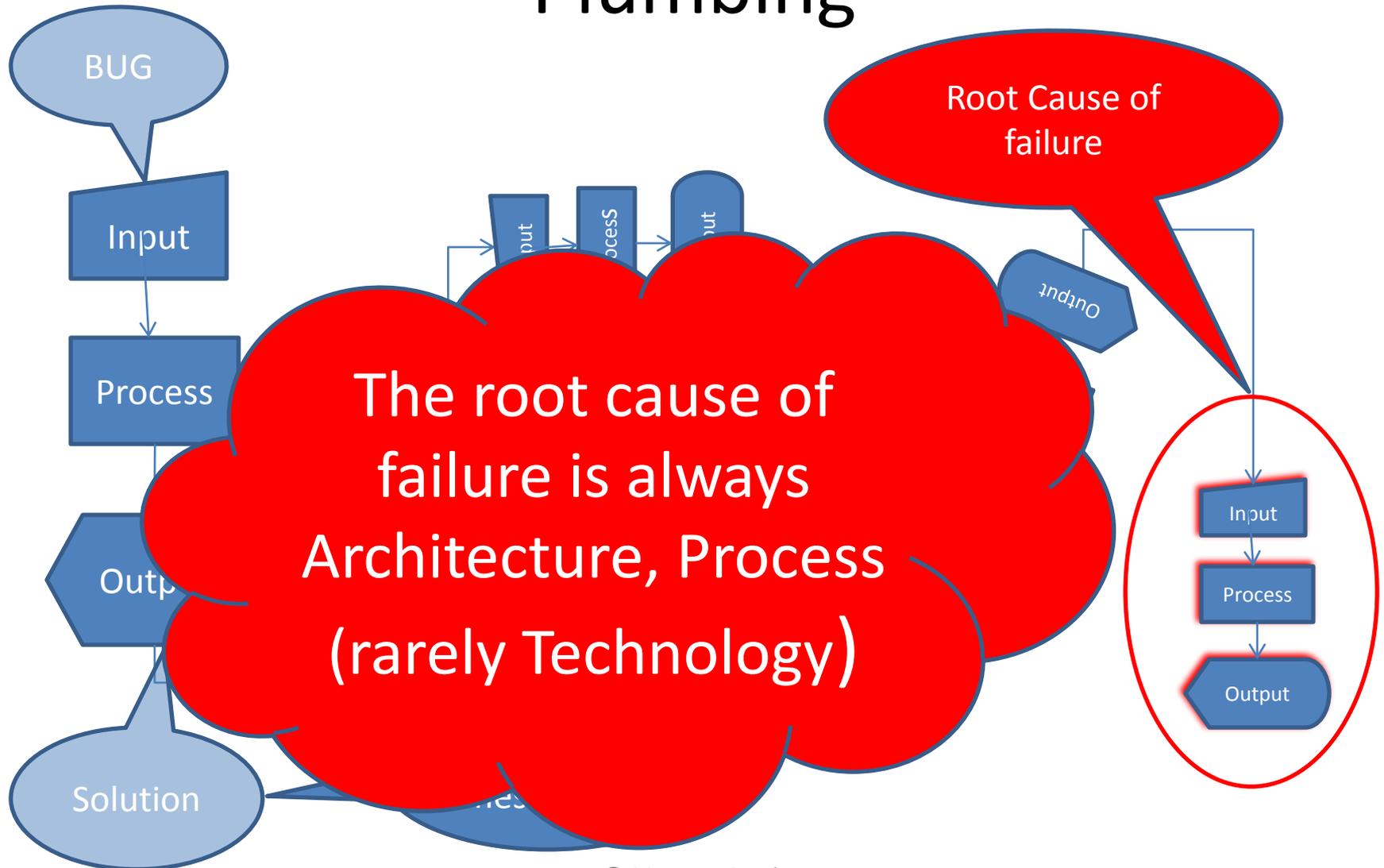
# About Gad J. Meir

- Experience: Since 1975
- Work: [www.idag.co.il](http://www.idag.co.il)
- Function: [www.productiondebugging.com](http://www.productiondebugging.com)
- Blog:  
<http://weblogs.asp.net/gadim/default.aspx>
- MSF Certified Trainer & Practitioner
- BSc. Computer engineering [Technion](#)
- Microsoft Certified MC...

# About IDAG Ltd.

- Founded 1983
- Established the first Microsoft certified training center in Israel at 1992.
- Areas of operation
  - Troubleshooting systems and procedures
  - Production time debugging to root cause of failure
  - Projects monitoring and guidance
  - Knowledge gaps detection and filling
  - Technologies and methodologies deployment

# From Bug Extermination to Process Plumbing



# I Have a Question 1/4

- Are you a
  - Developer?
  - Test/QA?
  - IT?
  - Management?
  - Other?



# I Have a Question 2/4

- Main Target Operating System
  - XP?
  - Vista?
  - Windows 7?
  - Server 2003?
  - Server 2008?
  - 2008 R2?
  - Other?



# I Have a Question 3/4

- Bit
  - 32?
  - 64?
  - Other?



# I Have a Question 4/4

- Run Time Environment
  - Managed (.NET)?
  - Native?
  - Other?



# Talk Targets

- Explain some of the specific constraints of production environment / Field
- Introduce ways to get debug data from production environment with minimum disruption to the System / Users
- Several scenario Demos for Native and Manage code
- Tips

# Prerequisites

- Experience in debugging

# Agenda

- Theoretical background (Quantum physics )
- What is a production environment
- Dumping bodies (AdPlus)
- Mapping the bodies (Symbols)
- Autopsying and analyzing bodies (WinDbg)
- The problem with the .NET way of handling bodies
- Tools for extracting information from .NET bodies (SOS)
- Things you can't get from a dead body
- Working with live bodies (Live Debugging)
- IIS (Debug Diag)
- Q & A

# Please !

- If you don't understand what I am talking about, Stop me and ASK !!!, Don't wait.

# Gad's Guidelines

- Nothing in life is certain
- If you measure it, it will be wrong
- Any action has at least one unexpected reaction
- Debugging application with Visual Studio, on a live production system, with 10,000 on line users, might affect your job security

## Theoretical Basis

[Uncertainty Principle](#): [Werner Karl Heisenberg](#) (1901-1976)

[Newton's Laws of Motion](#): [Isaac Newton](#) (1643-1727)

[Observer Effect](#)

[Murphy's Law](#)

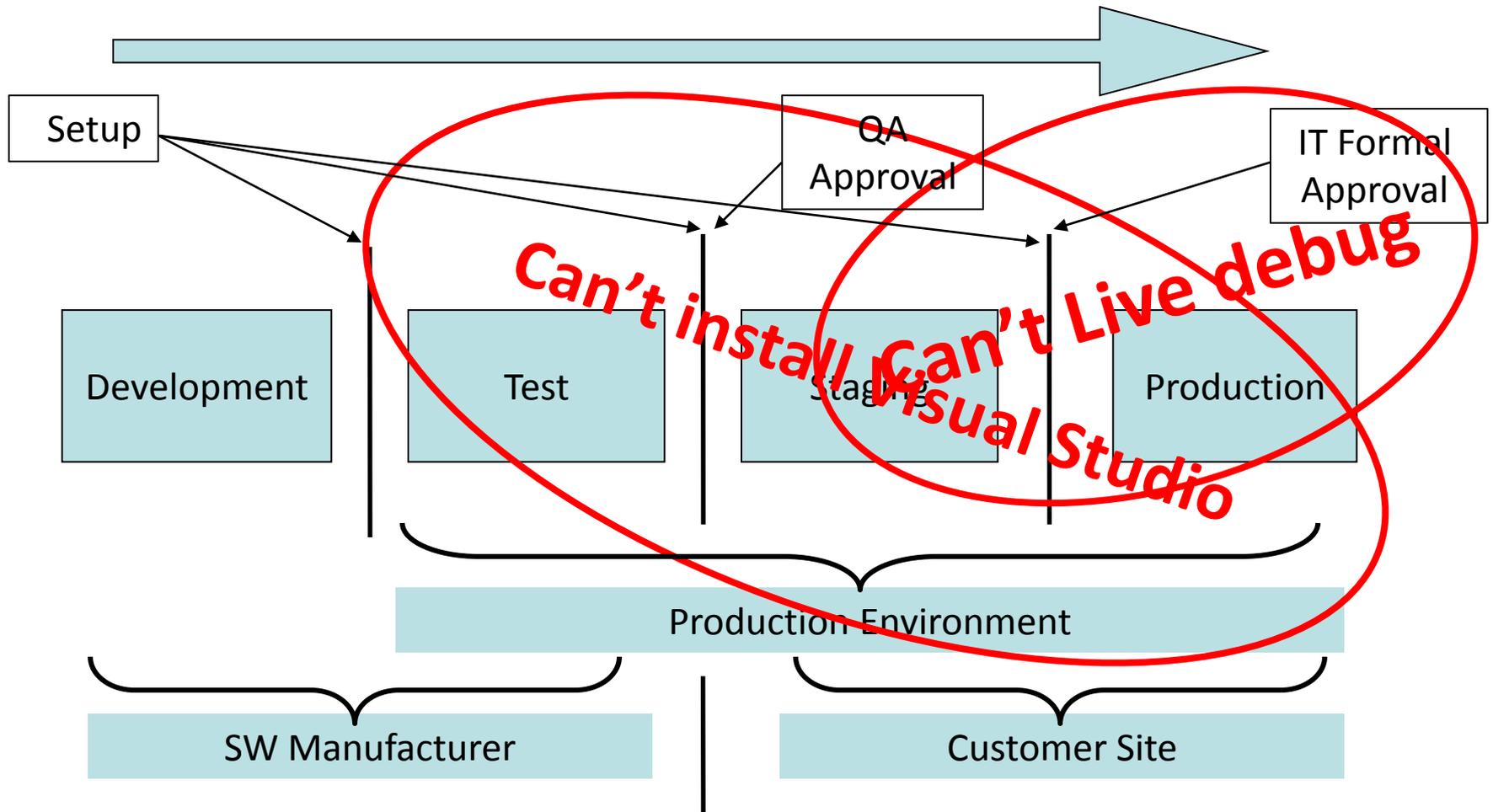
# What is a Production Environment



# What is a Production Environment

- Must be up and running all the time !!!
- Managed by administrators and help desk
- Under change control
- Managed remotely by management tools
- Different Hardware / Software
- Different OS constrains (Policy, Security, ...)

# Development & Production

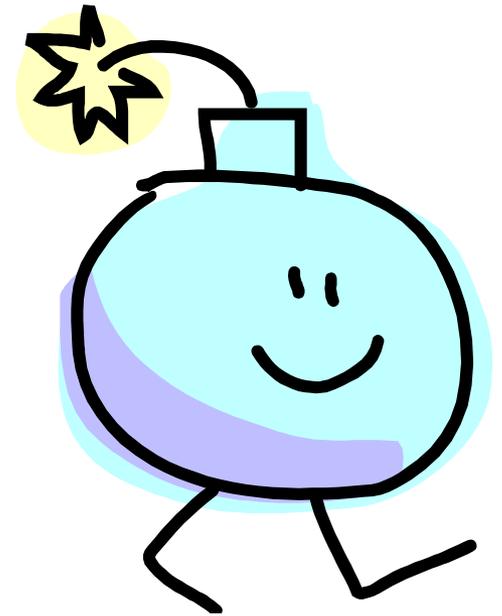


# About a Dump

- A snapshot of the process memory at the time you take the dump
- Easy to get in production environments with minimum intervention with the production
- In most of the cases includes all the information needed to analyze the problem

# Demo 010

- Analyzing a dump from a crashed program



# Pathology Basics

- A dead body is as good as a live one
  - The only thing you can't do with a dump is single-step it
  - You can duplicate and distribute dead bodies
- Conclusion and strategy # 1
  - Take the money and run

# 6 Easy Steps for beginners

- Get the tools
- Get the Symbols
- Set the environment
- Take a Dump
- Drop the dump into the tool
- !analyze

# How to Get the tools

- The Debugging tools for windows MSIs are In the SDK
- Download from <http://msdn.microsoft.com/windows/hardware> and go to Downloads
- Install once (for every hardware architecture)
- Zip and copy to you tools repository
- No need to install for using (Important for production)
- .

# How to Get the Symbols

- The Symbols MSIs are In the SDK
- Download from <http://msdn.microsoft.com/windows/hardware> and go to Downloads and than to Other hardware and development tools and than to Download windows symbol packages
- Install once (for every hardware architecture and OS)
- Put in a public location
- Remember the path

# Set the environment

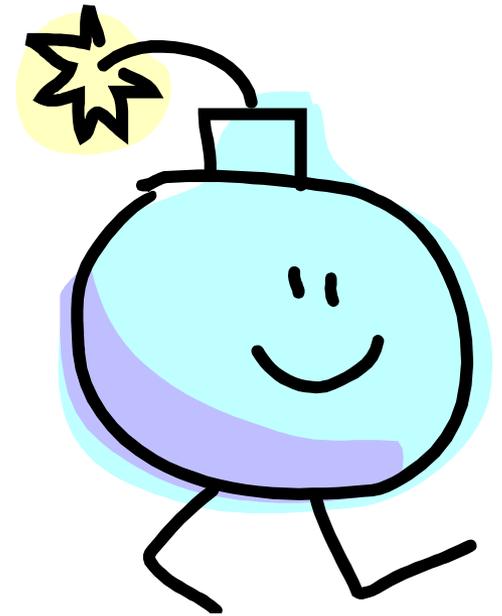
- Open WinDbg
- Set the symbol path
  - .sympath to app PDBs
  - .sympath+ to the Windows (correct version) PDBs
  - .symfix+ to the Microsoft Symbol server
- Save the WinDbg environment as a workspace for later use

# Tools to Take a Dump

- Adplus
- Windbg .dump
- Process Explorer
- Task Manager (Vista & Above)
- DebugDiag
- UserDump
- ProcDump
- WER
- ...

# Demo 020

- Taking a dump of a hanged program using Task manager

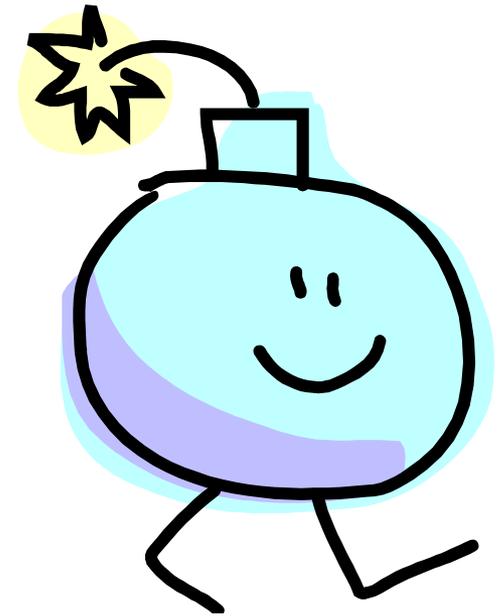


# About the different types of Dumps

- Application Mini dump
  - More or less just the call stack
- Application Full dump
  - Everything
- (Kernel dumps mini, kernel and full)
  - For BSODs

# Demo 030

- Taking a dump of a hanged program using WinDbg



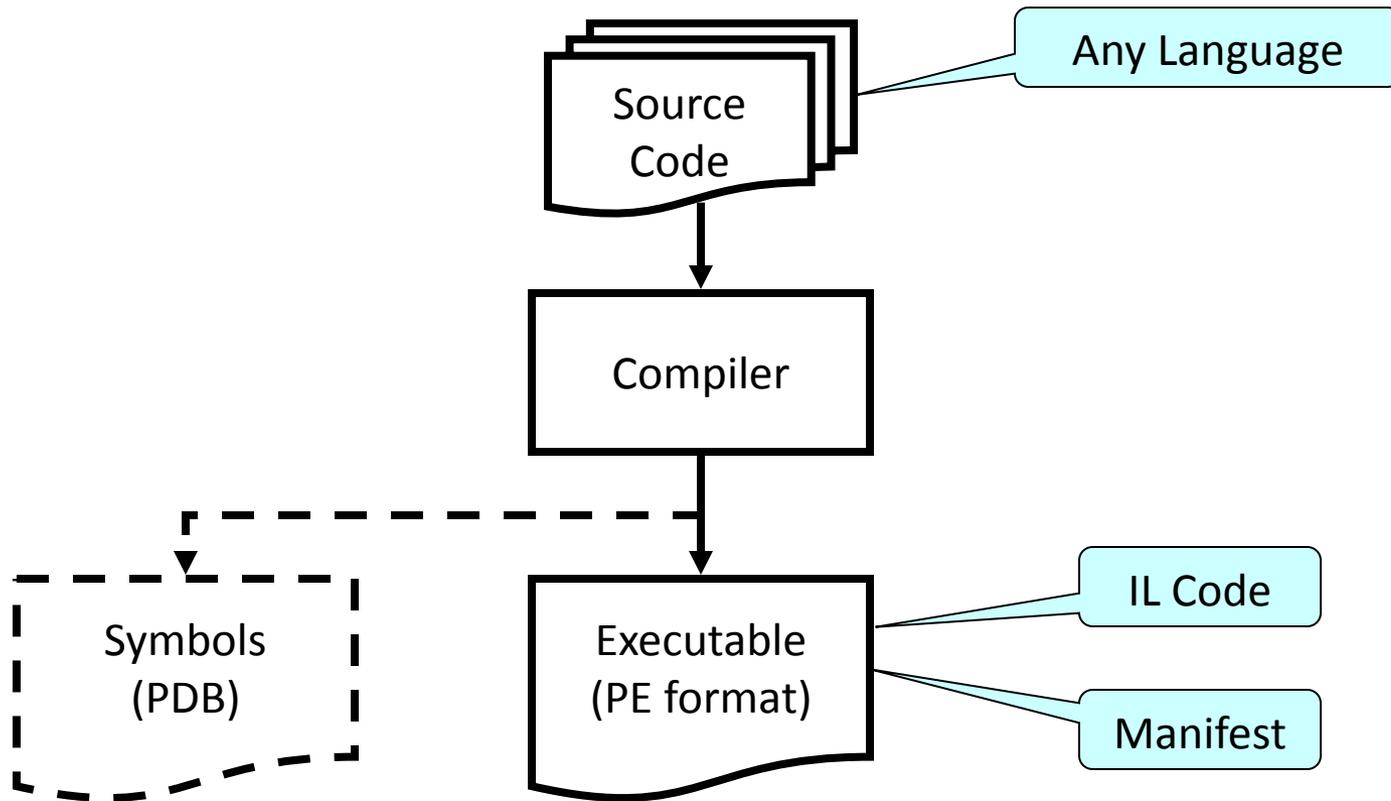
# About .NET (CLR)

- CLR is a win32 program!
  - A COM component
- CLR is the execution engine for IL code
- With win32 tools just the CLR engine is noticed
  - IL running code is ignored!
- SOS debugger extension is required
  - ‘Translates’ from Managed to Native

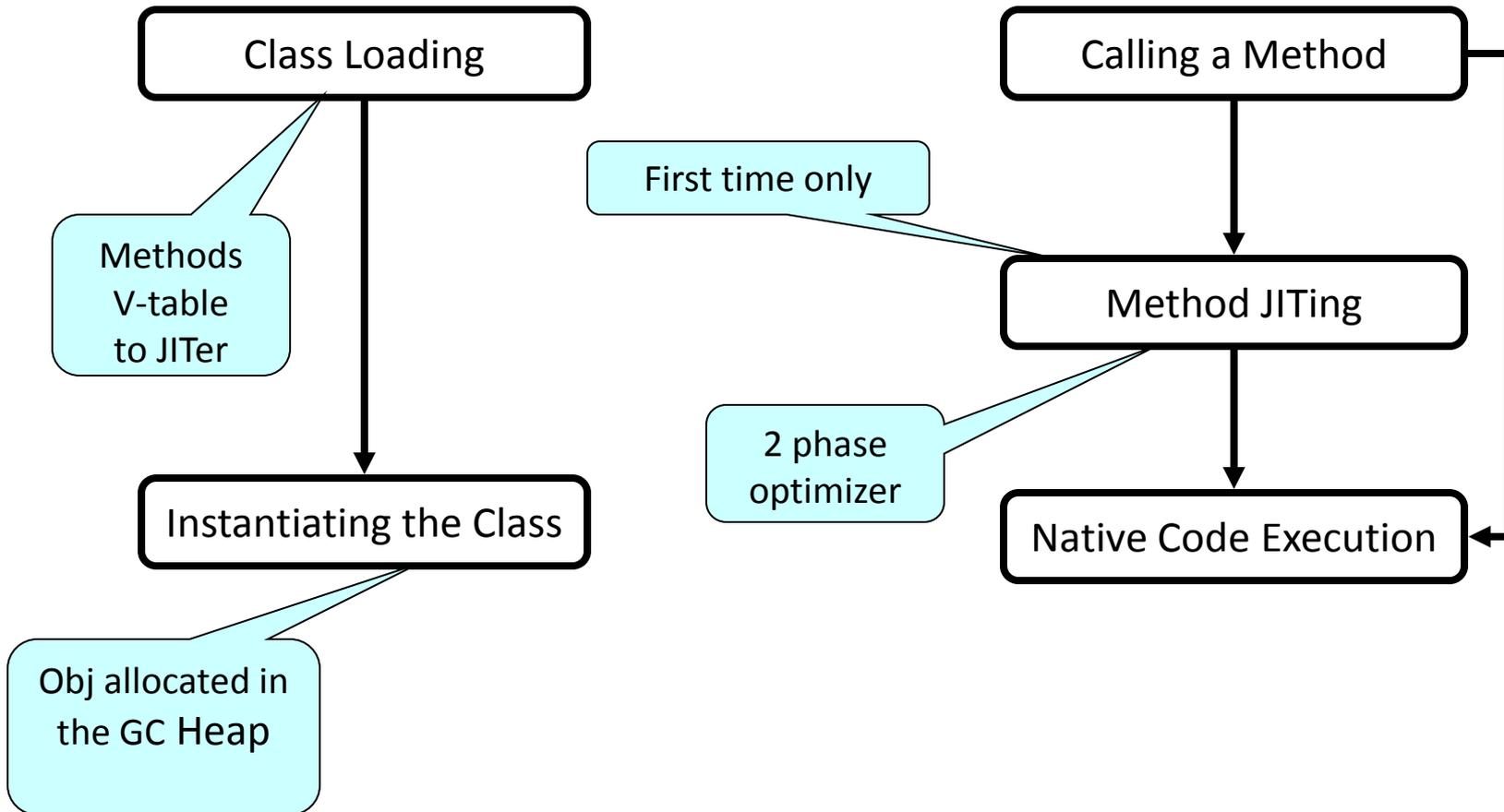
# Minimum .NET Internals

- Stack Machine (Reverse Polish Notation)
- Basic data unit is an Object
- The IL code is JITed into Native Code
  - On a function by function basis
  - On the first encounter

# Preparing the .NET Executable



# Running the Code in the CLR

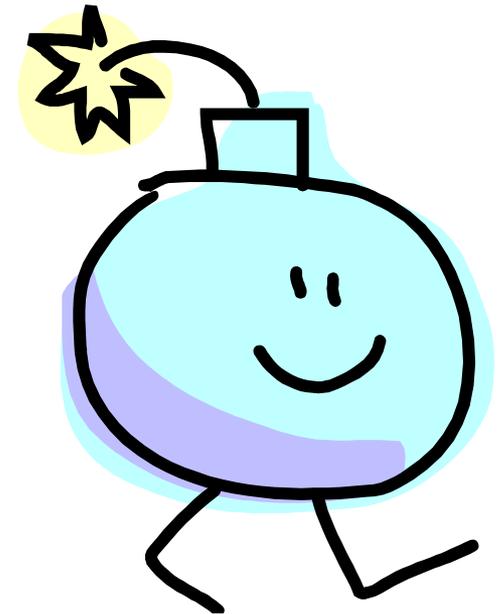


# Problems with .NET

- No PDBs for JITed code
- JITed code is 'nowhere'
- CLR handles all exceptions
- Hara-kiri effect when CLR can't handle an exception
  - By default, the CLR kills every one involved, cleans all the evidence from the crime scene and commits suicide, without leaving a comprehensible note

# Demo 040

- .NET Hara-kiri effect
  - Native Crash
  - Managed Crash

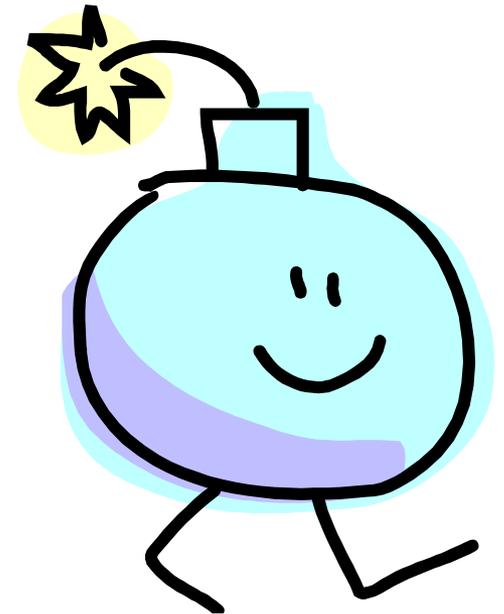


# SOS !Help

- **Object Inspection**
  - DumpObj (do)
  - DumpArray (da)
  - DumpStackObjects (dso)
  - DumpHeap
  - DumpVC
  - GCRoot
  - ObjSize
  - FinalizeQueue
  - PrintException (pe)
  - TraverseHeap
  - **Examining CLR data structures**
  - DumpDomain
  - EEHeap
  - Name2EE
  - SyncBlk
  - DumpMT
  - DumpClass
  - DumpMD
  - Token2EE
  - EEVersion
  - DumpModule
  - ThreadPool
  - DumpAssembly
  - DumpMethodSig
  - DumpRuntimeTypes
  - DumpSig
  - RCWCleanupList
  - DumpIL
- **Examining code and stacks**
  - Threads
  - CLRStack
  - IP2MD
  - U
  - DumpStack
  - EEStack
  - GCInfo
  - EHInfo
  - COMState
  - BPMD
  - **Diagnostic Utilities**
  - VerifyHeap
  - DumpLog
  - FindAppDomain
  - SaveModule
  - GCHandles
  - GCHandleLeaks
  - VMMap
  - VMStat
  - ProclInfo
  - StopOnException (soe)
  - MinidumpMode
  - **Other**
  - FAQ

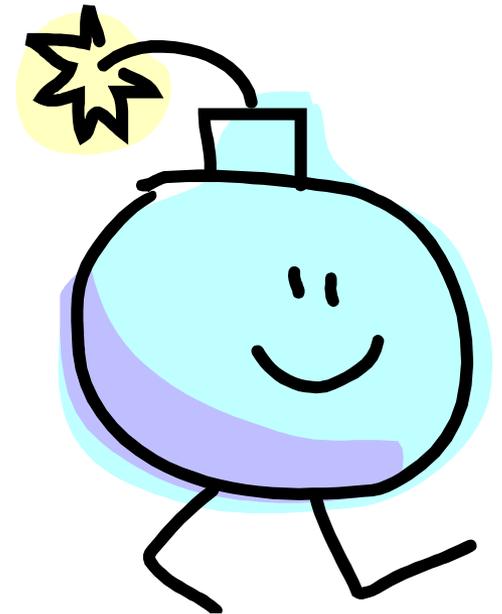
# Demo 050

- WinDbg Native and Managed view of .NET program
  - Without SOS
  - With SOS



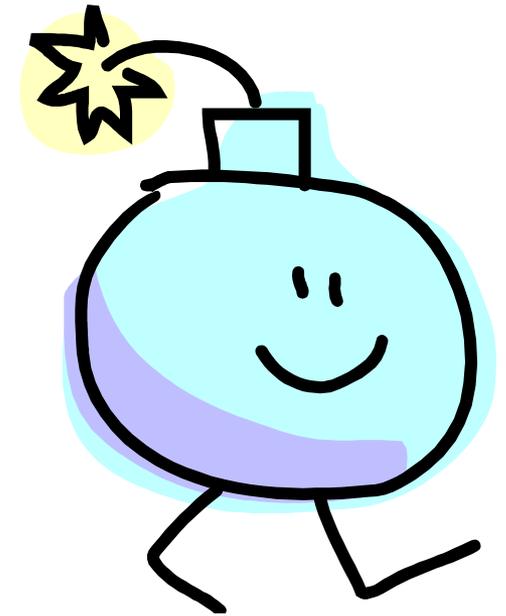
# Demo of a .NET Crash 060

- Call Stack
  - !clrstack
- Objects and Values
  - !do
- Object Stack
  - !dso



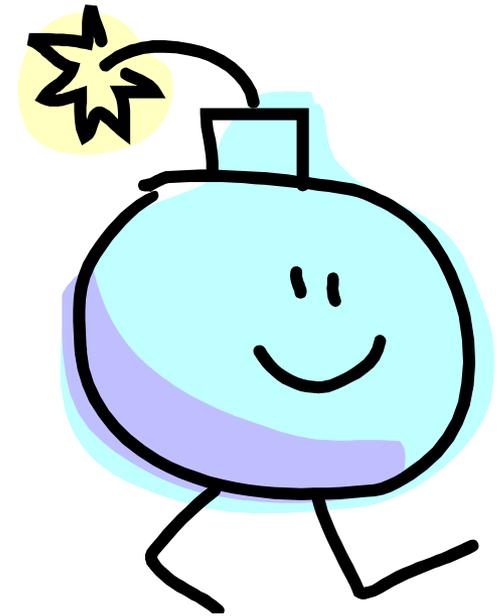
# Demo of a Deadlock Scenario 070

- !syncblk



# Demo of Finalization Starvation 080

- !finalizequeue



# Things You Can't Get From a Dump

- Dynamic behavior
- Leaking native memory
- Memory already corrupted before the dump
  
- Approaches
  - Time slicing / Event slicing
    - Scheduler
    - Performance Counter events
  - Live debugging

# Summery

- In the field you can't use the same techniques you use in development.
- Extracting dumps is one of the ways to gather information in the field without disturbing production.
- Instrumentation is key to help you gather information in the field

# If you want to learn more

- IDAG Ltd. have a 3 day of practical workshop on the subject of “Production Time debugging”.
- The workshop contain practical labs based on real live scenarios.
- The workshop includes all the methodology and practical consideration to properly debug application in the field.

# Resources

- <http://msdn.microsoft.com/windows/hardware>
- [winqual.microsoft.com](http://winqual.microsoft.com)
- “Debugging tools for Windows” help file
- “Debugging tools for Windows” SDK
- [Debugging MS .NET 2.0 Applications](#) Ch 6
- [MSDN patterns & practices Debugging](#) (Archived)
- !SOS.help & Q&A
- <http://blogs.msdn.com/tess>
- <http://support.microsoft.com/kb/q286350/>
- [\*\*Advanced Windows Debugging\*\*](#)
  - ISBN 0-321-37446-0 ,Addison Wesley, Mario Hewardt & Deniel Pravat

# Some Philosophy

- IT managers appreciate professionalism
  - Be prepared, know your tools and their footprints
  - Learn enough about IT to show them you are not the enemy
  - Listen, Listen, Listen
- Listen to the customer !
  - You developed it, but they use it every day
  - Write everything they complain about and put it straight into the product wish list

# Questions?



**Gad J. Meir**

**IDAG** Ltd.



**Bug Exterminator & Process Plumber**

**EBlog:** [weblogs.asp.net/gadim](http://weblogs.asp.net/gadim)

**HBlog:** [blogs.microsoft.co.il/blogs/gadim](http://blogs.microsoft.co.il/blogs/gadim)

**Email:** [gadim@idag.co.il](mailto:gadim@idag.co.il), **Site:** [www.idag.co.il](http://www.idag.co.il)



# Thank You!



**Gad J. Meir**

**IDAG** Ltd.



**Bug Exterminator & Process Plumber**

**EBlog:** [weblogs.asp.net/gadim](http://weblogs.asp.net/gadim)

**HBlog:** [blogs.microsoft.co.il/blogs/gadim](http://blogs.microsoft.co.il/blogs/gadim)

**Email:** [gadim@idag.co.il](mailto:gadim@idag.co.il), **Site:** [www.idag.co.il](http://www.idag.co.il)

**Copyright © 2011 by IDAG Ltd. and Gad J. Meir.  
All rights reserved. (Some parts quote  
Microsoft public materials). This presentation,  
its workshops, labs and related materials may  
not be distributed or used in any form or  
manner without prior written permission by  
the author(s).**