



AQI//A



Innovations in modern Accounting Software



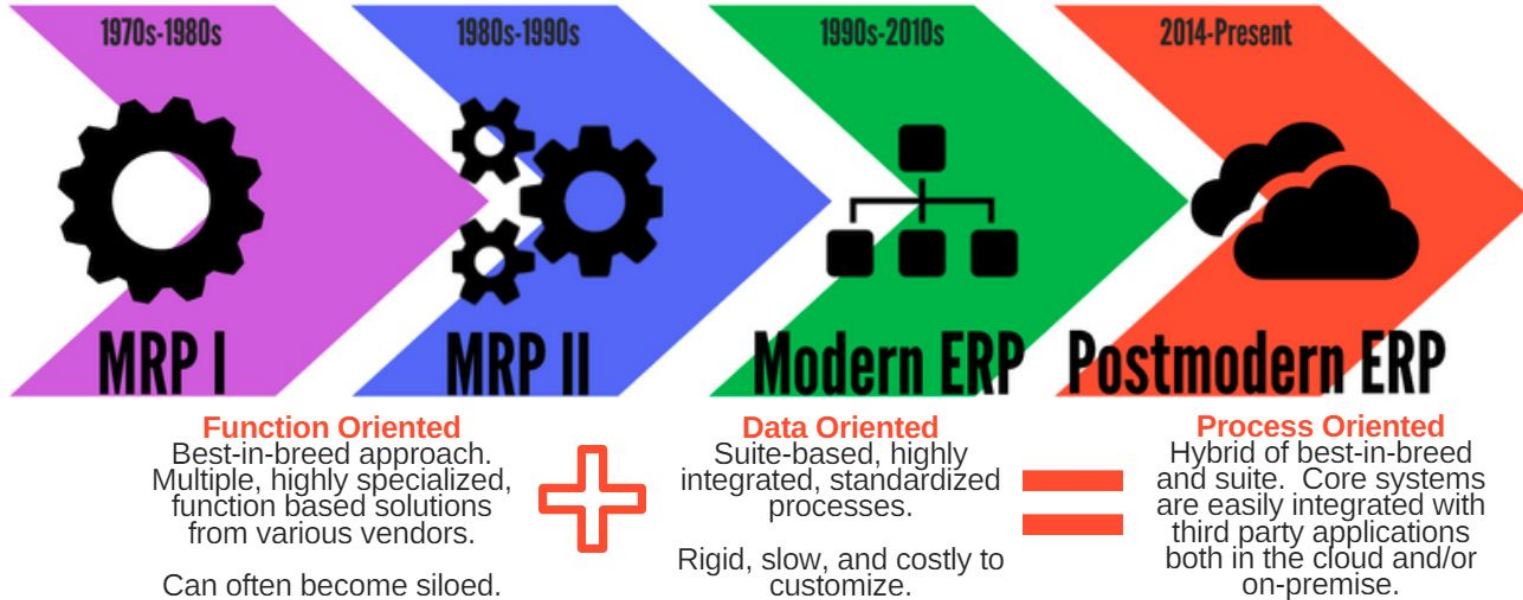
- Under pressure to provide greater reporting and insight to directors
- Need to implement efficiencies to support focus on analytics
- No longer functional - doing more outside your system than with it, probably time for a change !

GrowCFO Survey Results

- 66% of responders rated their software's sophistication lower than 7
- 25% of responders rated their finance system a 5 or less with respect to it how closely it meets their current requirements.
- 46% rated it as an 8 or above.



Why do you need a new system?



“A traditional ERP system is like the new car you buy every 10 years. A postmodern ERP system is like owning the same car indefinitely, but with various components that can easily be changed out as needed.”

The Evolution of Accounting Software

On-Premises

9%

Software Licenses

Customisation &
Implementation

Hardware

IT Personnel

Maintenance

Training

Ongoing Costs

- Apply Fixes, Patches, Upgrade
- Downtime
- Performance tuning
- Rewrite customizations
- Rewrite integrations
- Upgrade dependent applications
- Ongoing burden on IT
- Maintain/upgrade hardware
- Maintain/upgrade network
- Maintain/upgrade security
- Maintain/upgrade database

Cloud Computing

68%

Subscription Fee

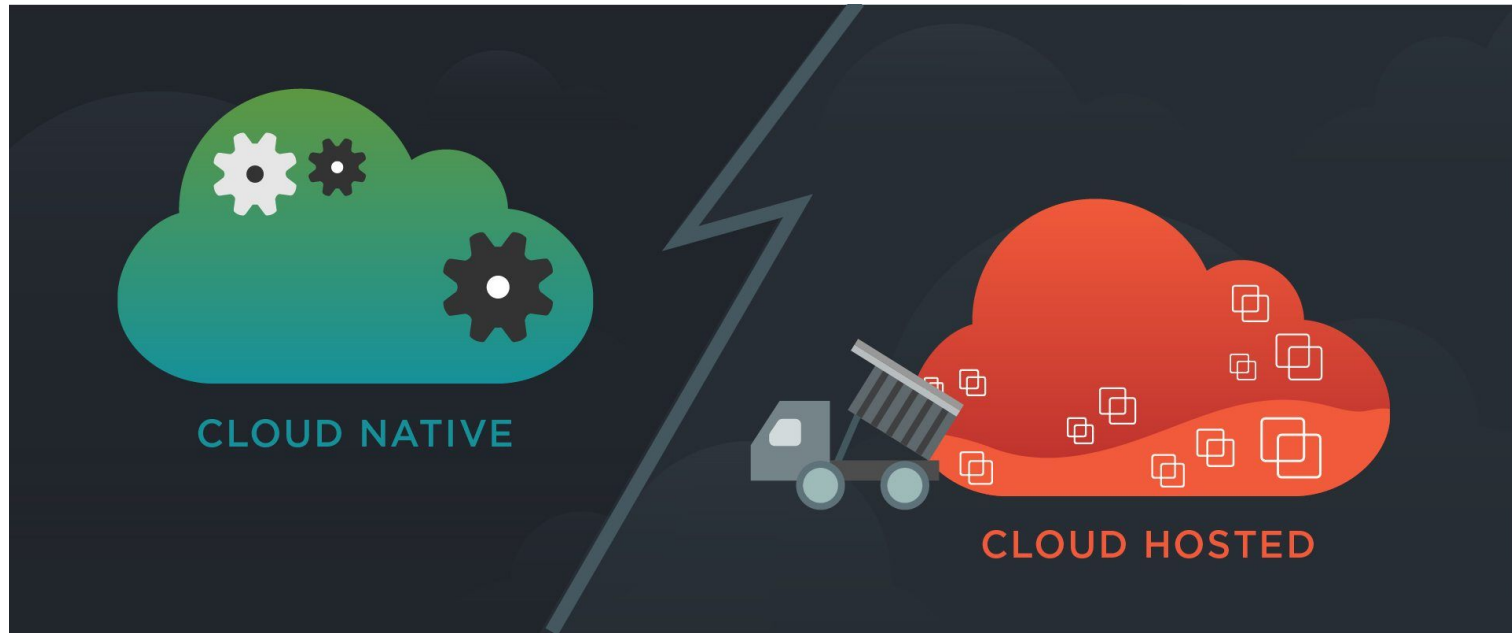
Implementation,
Customisation &
Training

Ongoing Costs

- Subscription fee

On Premise V Cloud

AQi//A



(<https://growcfo.mn.co/posts/cloud-native-vs-cloud-hosted-a-guide-through-the-storm>)

Cloud-Native vs Cloud-Hosted

Traditional Modular Ledger systems have separate General, Purchase and Sales Ledgers which reflect times when accountants wrote information into large paper books or ledgers. Balances on control accounts were copied from one book to another, so that a full set of accounts could be completed and as an additional process control accounts were reconciled. This ensured that all of the individual entries added correctly to the control totals before any transfers were made.

Modern Unified Ledger: "A Unified Ledger (aka single combined ledger) is a single book in which two entries, a debit and credit, are made at the same time. All of the debits add to the same value as all of the credits. The result is that no control accounts are required and no transfers to other ledger books."

(<https://growcfo.mn.co/posts/unified-ledgers-v-modular-ledgers-whats-the-difference>)

Key benefits of Modern Ledgers:

- Eliminates repetitive creation of multiple entries for every transaction
- Removes the need to reconcile individual ledgers
- Real Time posting to ledger provides instant access to the latest data
- Speeds up period / year-end closing
- No requirement to reconcile control accounts
- No opportunity for your system to be out of balance
- No hidden background processes within software that need to be checked or maintained

Modern Unified Ledger V Traditional Modular Ledger

System Architecture

Transactional based systems are far better equipped by design to manage volume processing unlike those built on event-based architecture (typically systems derived from CRM) that throttle processing speeds when faced with large volumes.



Transactional V Event based systems

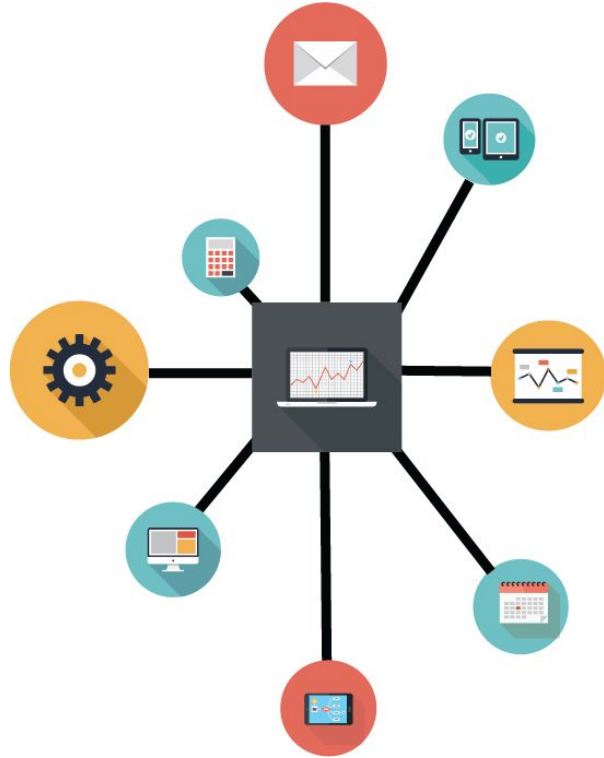
```

each: function(e, t, n) {
  var r, i = 0,
      o = e.length,
      a = M(e);
  if (n) {
    if (a) {
      for (; o > i; i++)
        if (r = t.apply(e[i], n), r === !1) break
    } else
      for (i in e)
        if (r = t.apply(e[i], n), r === !1) break
  } else if (a) {
    for (; o > i; i++)
      if (r = t.call(e[i], i, e[i]), r === !1) break
  } else
    for (i in e)
      if (r = t.call(e[i], i, e[i]), r === !1) break;
  return e
},
trim: b && !b.call("\uffeff\u00a0") ? function(e) {
  return null == e ? "" : b.call(e)
} : function(e) {
  return null == e ? "" : (e + "").replace(C, "")
},
makeArray: function(e, t) {
  var n = t || [];
  return null != e && (M(Object(e)) ? x.merge(n, "string" == typeof e ? [e] : e) : h.call(n, e)), n
},
isArray: function(e, t, n) {
  var r;
  if (t) {
    if (e) return e.call(t, e, n);
    for (r = t.length, n = n ? 0 > n ? Math.max(0, r + n) : n : 0; r > n; n++)
      if (n in t && t[n] === e) return n
  }
}

```

Low Code / Self Service Software

AQI//A



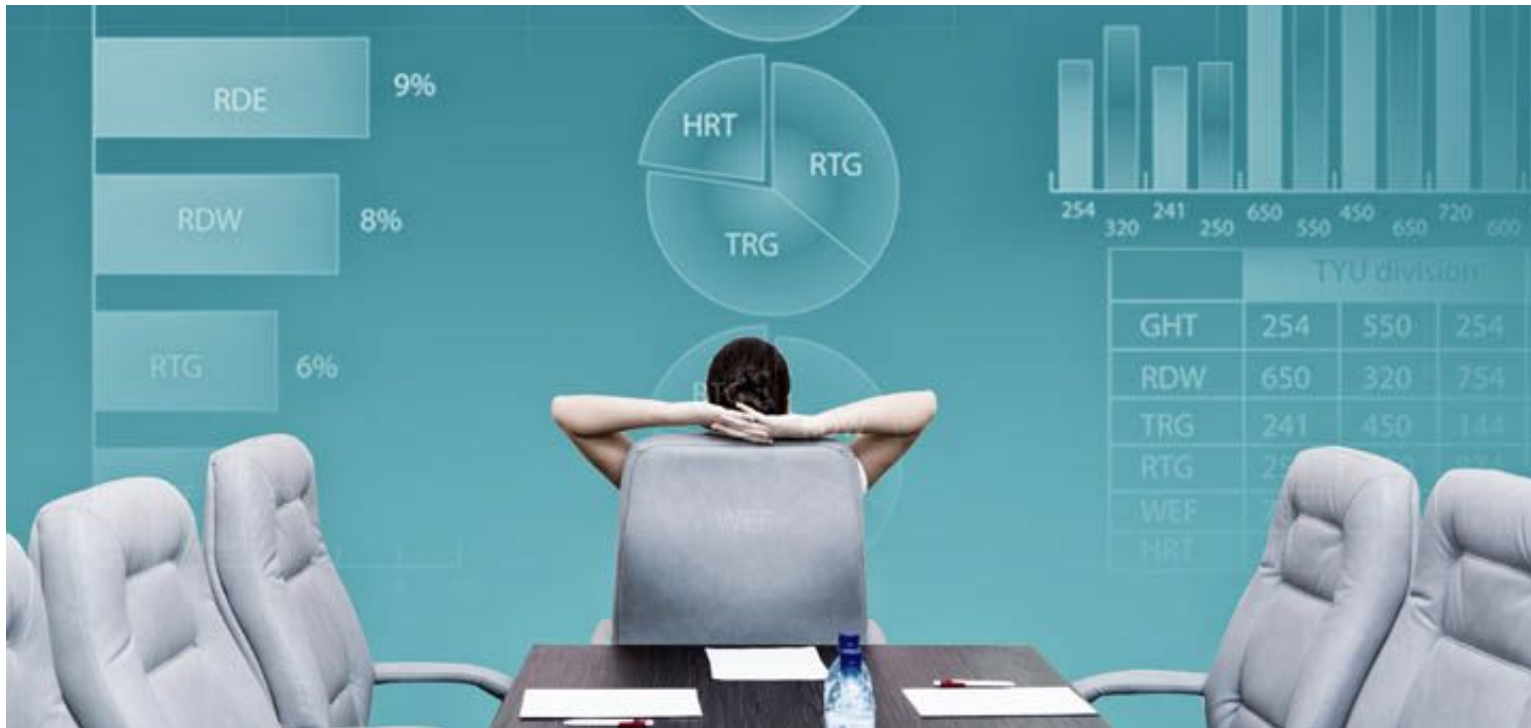
Agnostic integration



Automatic Maintenance & Updates



Real-time Data - Analysis V Insight



Extended Analysis

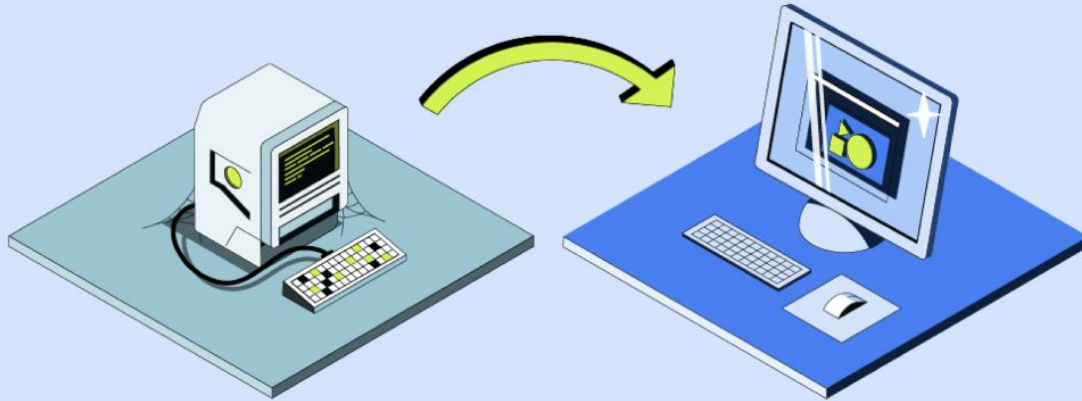
Functionality such as:

- Journal Mapping
- User Governance (Rules / Permissions)
- Workflows
- Reference Data



Empower the entire business

AQi//A



Simplified migration (Speed & Cost)



AQi//A